MacDroid:一种Android轻量级内核层强制访问控制框架

李尼格 马媛媛 陈 牧 陈 璐 徐 敏
(全球能源互联网研究院信息通信研究所 南京 210003)

摘 要 智能移动终端已成为移动互联网时代重要的信息处理平台,其面临的安全威胁越来越严重,针对传统计算机的安全防护架构已无法适应智能移动终端安全防护的特殊需求。通过对智能移动终端操作系统的特点和层次进行分析,设计了一种轻量级内核层强制访问控制框架——MacDroid,深入研究了MacDroid的安全策略定义、安全策略编译、安全策略实施等关键问题。提出了MacDroid的安全策略描述语言——PSL,对PSL的词法和语法进行了形式化定义。最后通过实验测试了MacDroid访问控制框架对智能移动终端不同层的恶意软件行为的控制效果。实验结果表明,MacDroid框架对Android智能移动终端应用层、本地层和内核层的恶意软件行为均有较好的控制效果。

关键词 安卓,内核,强制访问控制,恶意软件检测
中图法分类号 TP309 **文献标识码** A

MacDroid:A Lightweight Kernel-level Mandatory Access Control Framework for Android

LI Ni-ge MA Yuan-yuan CHEN Mu CHEN Lu XU Min
(Institute of Information and Communication, Global Energy Interconnection Research Institute, Nanjing 210003, China)

Abstract Smart terminal has become an important information processing platform in the mobile Internet era, and its security threats are becoming more and more serious. The security protection architecture for traditional computers has been unable to meet the special needs of smart terminal security protection. By analyzing the characteristics and levels of the smart terminal operating system, a lightweight kernel-level mandatory access control framework(MacDroid) was designed. The key issues of MacDroid security policy definition, security policy compilation, security policy implementation and so on were deeply studied in this paper. The MacDroid security policy description language(PSL) was proposed and the PSL lexical and grammar formal definition were given. Finally, the effect of MacDroid access control framework on the behavior of different layers of intelligent mobile terminals was evaluted. The experimental results show that the MacDroid framework has good control effect on application layer, native layer and kernel layer malware behavior of Android smart terminal.

Keywords Android, Kernel, Mandatory access control, Malware detection

1 引言

传统的操作系统保护依赖于安全策略,系统安全管理员通过分析应用程序的安全策略可决定应用程序在运行时的权限。智能移动终端相比于传统的桌面型计算机在信息获取、信息传输方面的优势更加明显,它们通常可更方便地访问网络并且更便携,但是它们的计算能力和存储能力与桌面型计算机相比还存在一定差距;同时,它们的电量是受限的。这些新的特征使得基于传统的桌面计算机研究的安全防护技术并不完全适用于智能移动终端,必须为智能移动终端研究新型的安全防护技术。

访问控制技术是信息系统安全防护研究的重要内容[1]。当前关于访问控制技术的研究主要关注访问控制模型、访问控制策略设计、访问控制决策算法等,传统的桌面型计算机研究已经取得了许多成熟和稳定的成果,包括访问控制算法和实现技术等;但是针对智能移动终端等计算、存储和电量受限的新型嵌入式设备的访问控制技术研究还相对较少。智能终端上运行的软件通常不直接访问OS提供的API,它运行在一个虚拟层上面,只访问虚拟层的API,如IOS的Cocoa Touch层、Android的Framework层。这些特征的存在使得传统的访问控制模型和技术并不能直接应用于智能移动终端。

本文设计了一个Android轻量级访问控制模型,并在Android系统上实现并验证了该模型的有效性。该模型是在内核层实现的,后文将讨论在内核层实现访问控制的必要性和优势。测试表明,该模型可较好地防护内核层的Rootkit、恶意软件对硬件的恶意访问以及对文件系统的恶意破坏。

本文第2节讨论智能移动终端的访问控制、内核防护相关的研究工作;第3节设计并实现Android内核层强制访问控制模型MacDroid;第4节通过实验验证了该模型的有效性,并对模型的不足以及改进、设想进行了讨论。本文设计的

本文受国家电网公司科技项目:电力移动应用信息安全防护关键技术研究(SGRIXTKJ[2016]183号)资助。
李尼格(1985-),女,硕士,工程师,主要研究方向为电力系统信息安全,E-mail:linige@geiri.sgcc.com.cn;**马媛媛**(1978-),女,硕士,高级工程师,主要研究方向为电力系统信息安全;**陈 牧**(1986-),男,硕士,工程师,主要研究方向为电力系统信息安全;**陈 璐**(1984-),女,硕士,工程师,主要研究方向为电力系统信息安全;**徐 敏**(1984-),男,硕士,工程师,主要研究方向为电力系统信息安全。

访问控制模型不仅可以用于基于 Android 的智能移动终端，而且可以用于其他源码开放的智能移动终端操作系统。

2 相关工作

与传统的信息系统访问控制研究相比，针对智能移动终端的访问控制研究正处于“方兴未艾”的阶段。近年来，随着云计算、移动互联网和智能移动终端技术的飞速发展，针对智能移动终端的访问控制技术研究也逐渐展开。

Ion 等人[1-2]于 2007 年为通过扩展移动设备的 Java 虚拟机设计了一种细粒度(fine-grained)的安全策略。文中的方法适用于采用了 Java 虚拟机的多种类型的移动设备，但未考虑多层的移动设备软件架构的特点，不能解决本地层(native)和内核层的权限滥用问题。

Zhan 等人[3]于 2007 年为基于 Linux 的嵌入式系统设计了一种基于安全内核(secure kernel)的访问控制参考结构。该结构未考虑目前基于 Linux 的智能移动终端操作系统中大量存在 Java 类格式的可执行程序和各种类型的可执行脚本程序，无法对这几种常见的可执行程序进行控制，因此在对 Android 和 Motavista 等基于 Linux 的移动终端上的可执行程序进行访问控制时存在不足。

2009 年，Enck 等人[4]在研究了 Android 的权限机制和软件包安装机制后，发现 Android 的权限机制存在控制粒度不够、用户无法理解和配置的问题，并对此设计了一个增强型的程序安装器(installer)——Kirin。Kirin 可依据 Android 的 apk 安装包中的 AndroidMenifest 文件解析出该应用声明的权限，同时 kirin 还定义了一些危险的权限组成集合，如读取短信记录和发送邮件、获取地址和发送短信等，若应用声明的权限包含危险权限集合，则提示用户是否继续安装该程序。Kirin 可较好地解决声明危险权限组合的恶意应用和被重新打包过的(repacked)正常应用对系统安全造成的危害问题。该模型同样不能解决本地层恶意程序和内核层 Rootkit 对系统安全造成的威胁。

Kirkpatrick 等人[5]于 2010 年分析了移动设备(mobile devices)的特点后对传统的 RBAC 模型进行了扩展，提出了针对移动设备的 RBAC 模型。该模型使用部分限制(spatial constraint)代替全局限制。该文中的模型虽然能解决移动设备软件结构复杂、跨越层次多、访问控制模型实现困难等问题，但是将全局限制修改为局部限制后，模型本身实现的复杂度也成倍增加。

Nauman 等人[6]于 2010 年设计了一个针对移动终端平台的高效完整性度量模型，该模型可对从 bootloader 开始到软件运行的全过程进行完整性度量。其通过修改 bootloader 来对内核映像进行完整性度量，考虑了将 OS 中多种不同类型的操作归纳为读、写和创建；定义了进程间交互的完整性度量规则。该模型的不足在于：1)仅仅通过考虑进程的文件来自于只读分区或读写分区来对进程完整性进行度量，未考虑文件来源于何种途径，如蓝牙、红外、网络下载、软件市场在线安装等；2)模型未考虑进程的功能，如果通过 permission 等方式分析进程的功能，那么将其一并加入完整性度量模型将更加高效；3)完整性度量规则是静态的，无法通过运行过程中的知识积累进行动态调整。

2012 年，美国国家安全局 NSA 发布了 SEAndroid 源码[7]。SEAndroid 是一个庞大且复杂的安全增强型 Android 系统，它为 Android 系统建立类似于沙箱的隔离机制，以确保每一个 APP 独立运行。SEAndroid 具有成熟的安全模型和完善的安全策略，但是无法通过网络灵活且方便地更新安全策略。

针对 Android 安全机制存在的缺陷，黄琳雅[8]于 2012 年提出了一个 Android 文件访问控制系统，旨在为 Android 设备中存储的敏感文件提供访问控制机制。该方法无法对 Android 应用程序的行为进行控制，也无法方便地更新安全策略。

易筱茂[9]于 2015 年设计和实现了一种基于强制访问控制的应用隔离方案。该方案在 Android 现有安全机制的基础上进行扩展，针对 Android 中的各种访问方式实现内核层和框架层联动的完整强制访问控制。该方案需要修改 Android 框架层的源码，无法快速适配多种 Android 版本。

卿斯汉[10]于 2016 年阐述了 Android 安全的研究现状与发展趋势以及今后可能的研究方向。该文指出针对 Dalvik 虚拟机涉及的安全问题目前已有大量深入的研究，但鲜见关于 Android 5.0 版本后新引入的 ART 虚拟机的安全性研究，需要研究与 Android 上层运行环境机制无关的底层安全防护技术。

上述所有研究成果都无法解决 Android 和 IOS 等多层架构移动设备软件系统中各层权限的滥用问题，而内核是移动设备操作系统中最底层、最接近硬件的部分。研究智能移动终端操作系统内核层的安全防护技术，针对智能移动终端特点设计内核层访问控制模型，将从根本上解决目前智能移动终端设备面临的诸多安全问题，提高其安全性。

3 轻量级内核层访问控制框架

智能移动终端现有的安全框架通过两种方式来限制权限分配到应用程序：用户安装时确认(典型代表是 Android)和软件运行时确认(典型代表是 Apple IOS)。一款终端一旦发布后，其可信计算基应保持不变，且必须加以保护。应用程序的签名可在一定程度上为其提供对特别危险功能调用的防护，然而签名并不是绝对可靠的防护方式，应用程序的签名是可以伪造的，一旦一个应用程序伪造了签名并访问了终端上危险的功能，比如访问通信录、访问位置信息、发送短信等，将对终端的安全带来致命的威胁。

MacDroid 为智能移动终端软件签名验证机制和现有的访问控制机制提供了补充，可在操作系统底层对应用程序的行为进行控制。在普通智能移动终端中，每个应用程序通常都有一个对应的安全策略。操作系统根据其策略会在应用程序安装时或运行时对应用程序的行为权限进行判决，如果判决通过，则允许应用程序使用声明的权限对应的功能。实际上，这种访问控制模式存在很大的不足。

(1)应用程序声明多余的权限。正常情况下，应用程序只需要申请其需要的最小权限即可，如果申请多余的权限，就有实施恶意破坏行为的可能。比如，一款记事本应用申请了读取通信录和访问网络权限，那么它就有可能读取本地通信录后将其发送到攻击者的邮箱中。

(2)应用程序正常申请权限，但是滥用权限。正常情况下，应用程序应该将其申请的权限在正常范围内使用，但是有

可能存在正常申请的权限被滥用的问题，比如，一款应用程序申请了访问网络的权限，那么它可以访问网络，同时它也可能利用访问网络的权限从网络下载恶意程序到本地安装。

目前的智能终端访问控制机制无法解决上述两方面的问题，但 MacDroid 框架可较好地解决上述两方面的问题。图 1 给出了 MacDroid 框架中应用程序的安装和运行流程。其中，左边部分是应用程序的安装流程。在 MacDroid 框架中，应用程序的安装过程和安装包是经过改造的，安装程序包括两部分：改造过的安装包（Refactored Package）和安装包签名（Package Signature）。改造过的安装程序包括 3 部分：原始安装包（Original Package）、安装脚本（Install Policy）和内核层访问控制策略（Kernel Access Control Policy）。签名用来验证改造过的安装包的完整性。图 1 右边部分是应用程序的运行流程。应用程序在运行时要通过操作系统原有的权限检查和 MacDroid 框架中的内核层权限检查，只有两项检查都通过后应用程序才能正常运行。

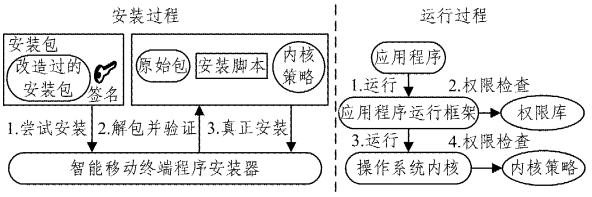

图 1　MacDroid 框架工作流程图

在 MacDroid 框架中，安装应用程序时，智能移动终端上配置的经过改造的程序安装器首先会读取签名并检查该安装包的完整性，如果签名验证不通过，则说明该包是经过篡改的安装包，直接退出安装过程；如果签名验证通过，则进行软件包的解压，分析安装脚本并开始安装程序，安装完成后将包中的内核层访问控制策略加入内核。应用程序运行时，由应用程序运行框架和操作系统内核分别对程序的运行权限进行检查，检查通过后程序开始运行。

3.1　MacDroid 安全策略

在前面的章节中介绍了智能移动终端面临的恶意软件威胁和其软件架构，这些都是本节的研究基础。安全策略是安全防护模型的重要组成部分，安全策略主要用来描述主体对客体的访问规则。下面将详细分析智能移动终端内核层的安全防护策略，以及这些策略在 MacDroid 中的设计与实现。

3.1.1　安全策略的类型

通过上文的安全威胁分析可以知道，内核是操作系统最核心的部分，一旦它被攻击和破坏，将会对系统安全带来最严重的威胁。为了在内核层对操作系统进行安全防护，首先需要制定一系列的安全策略，安全策略是安全防护的最主要组成部分。

智能移动终端的一个重要特点是允许终端用户在应用发布后通过安装软件来扩展系统的功能。用户在安装软件时通常无法判断软件的来源和可靠性，因此修改软件的安装流程，在安装过程中加入可靠性验证和安全策略的加载，是一种可行的方案。由于智能移动终端网络接入的便利性，通过网络更新终端的操作系统和软件也是软件安装的一个特点，因此通过网络下发的方式实时更新终端上的访问控制策略也是一种针对智能移动终端的有效的访问控制策略维护方式。

在 MacDroid 中，将安全策略分为两大类：系统预定义策略（system pre-install policy）和实时下发策略（Realtime dispatch policy）。

系统预定义策略是指在软件安装的同时加载针对该软件的内核层访问控制策略。通过修改智能移动终端的软件安装流程，可以限制智能终端只能安装经过定制的特殊格式的软件，其他格式的软件会被系统自动拒绝安装。在这种定制的软件安装包中含有针对该软件的内核层访问控制策略，这类策略会在软件安装完成后持久化保存并加载到系统内核中。系统预定义策略的执行流程如图 2 所示。

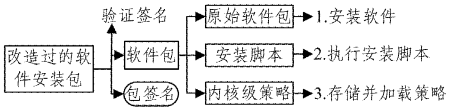

图 2　系统预定义策略的执行流程图

实时下发策略是指在系统运行期间，由网络侧的安全管理服务器通过 OTA（Over The Air）等方式向终端下发专用的内核层安全控制策略，终端接收到策略后进行解析、存储并加载到系统内核中。实时下发策略的执行流程如图 3 所示。

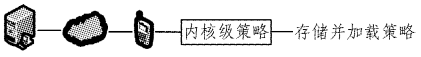

图 3　实时下发策略的执行流程图

3.1.2　策略的编辑与编译

智能移动终端内核层的访问控制策略需要人工编写，但是人工编写的文本格式的安全策略文件并不能直接被加载到内核中执行，必须对文本格式的策略进行转换生成机器可识别的安全策略后再将其加载到内核中执行。下面给出策略编译（Policy Compile）的形式化定义。

定义 1（安全策略编译，Security Policy Compile，SPC）假设文本格式的安全防护策略集合为 Pt，pt 为 Pt 中的一个策略，pt 满足如下条件：

$$\forall\, pt \in Pt \wedge Recognize(person, pt) \wedge \neg Recognize(k, pt)$$

其中，$person$ 表示人工，k 表示操作系统内核。假设机器可识别的安全防护策略集合为 Pb，pb 为 Pb 中的一个策略，pb 满足如下条件：

$$\forall\, pb \in Pb \wedge \neg Recognize(person, pb) \wedge Recognize(k, pb)$$

那么，安全策略编译 SPC 是一个函数 $ftranslate: pt \rightarrow pb$。

从 SPC 的定义可以知道：SPC 的过程就是将人工编写的内核不可识别的安全策略“翻译”为内核可识别的安全策略。在安全策略编译过程中，编译程序首先要检查策略的要素是否齐全，如果策略要素不齐全，则直接退出编译过程；然后遍历文本格式的策略文件，分别将主体安全策略和客体安全策略翻译成内核可识别的安全策略，在翻译过程中如果发现主体和客体相同，则报错。图 4 给出了安全策略编译的算法。

```
algorithm 1:PolicyCompile
1. if (!contain(subjects policy) && !(objects policy)) exit;
2. for (read subjects & objects policy) {
3.      pb(s,i)←pt(s,i);
4.      pb(o,i)←pt(o,i);
5.      if (S(j)==S(i) || (O(j)==O(i))) give error rowID;}
```

图 4　安全策略编译 SPC 算法

3.2　MacDroid 安全描述语言

为了形式化描述 MacDroid 安全策略，定义了 MacDroid

安全规则描述语言(MacDroid Security Language,PSL)来对MacDroid中所有的安全策略进行形式化描述。形式化语言描述安全策略的优点在于:

1)可方便地查找和发现安全策略的错误。真实的智能移动终端系统中的软件非常多,功能复杂,对应的内核层的安全策略也非常繁杂,涉及的主体和客体资源种类繁多,在编写安全策略时极易出错。使用形式化的描述方法可有效地对策略的"一致性"进行分析,防止出现错误的策略和自相矛盾的策略。

2)有利于安全策略编译工具的设计和实现。安全策略的编译工具与一般编译器的作用类似,都是将文本指令翻译为机器指令,只有严格形式化定义策略描述语言的语法和语义后,才能实现安全策略的自动化翻译。

本节形式化定义PSL的语法和语义。

3.2.1 PSL 词法

图5中使用BNF符号定义了PSL的词法。一个PSL策略集包含一个策略列表,策略列表由若干条规则组成。策略表示组合策略类型标签、类型和对象类型。每个规则以关键字"module"开始,它的值可以是"module"或"global",分别代表模块策略和全局策略。策略的其余部分是对象和操作字符串的组合。策略类型包括"允许策略"和"禁止策略",在安全策略的〈policy-type〉字段中使用"allow"和"deny"标识。

```
〈module〉module〈/module〉
〈policy-type〉allow〈/policy-type〉
〈policy-timeline-start〉2012-01-01〈/policy-timeline-start〉
〈policy-timeline-end〉2012-12-31〈/policy-timeline-end〉
〈policy-segment-identifier〉subject〈/policy-segment-identifier〉
〈full-path-name〉/data/com. test. test〈/full-path-name〉
〈subject-type〉SUBJECT_JAVA〈/ subject-type 〉
〈policy-segment-identifier〉object〈/policy-segment-identifier〉
〈full-path-name〉/data/dara/com. test. test〈/full-path-name〉
〈object-type〉OBJECT_DIR〈/ object-type〉
```

图5 使用BNF定义的PSL词法

全局策略(Global Policy):在访问控制决策过程中没有任何安全策略的情况下使用的策略。

模块策略(Module Policy)文件中包含零个或多个主体的安全政策和零到多个对象的安全策略。每条主体策略包含主体对应的进程的全路径和主体类型等信息,每条客体策略包含客体全路径和客体类型等信息。

在定义了PSL的词法后,就可以使用PSL来为MacDroid描述各种内核层的访问控制策略。

3.2.2 PSL 语法

定义了一个简单的逻辑来表示使用PSL表示的一组策略。假设RA表示PSL的所有策略。设M表示策略类型的集合,S表示可能的访问主体类型的集合,O表示可能的访问客体类型的集合,sf表示主体的全路径,of表示客体的全路径,则任意一条策略$r_i \in RA$是一个n元组:$(m, si, sf, st, 2^{si,of,ot,p})$,其中$m \in M$, $si \in SI$, $st \in ST$, $ot \in OT$, $p \in P$。$2^{si,of,ot,p}$表示集合(si,of,ot,p)的幂集,含义是策略r_i可以包含一条或多条客体访问规则。

定义了MacDroid的语义后,下面讨论策略的生成过程。对于任意一个安装包,依据信息安全中的最小特权原则,首先需要明确其正常运行所需要的最小权限,假设C是安装包中声明权限的集合,RA是PSL描述的内核层策略集合,那么需要找到一种映射$FR: C \to RA$,将应用程序安装包中声明的权限转化为使用PSL描述的内核层访问控制权限。

现在就可以定义PSL策略集的语义。函数$fail$定义如下:

$fail: C \times R \to \{success, fail\}$

$fail$函数用来测试是否成功地将软件安装包中定义的权限集C转化为PSL描述的策略R。假设c_t表示从目标软件安装包t中获取的权限集合,r_i表示针对t使用PSL描述的策略,定义$fail(c_t, r_i)$表示将c_t转化为r_i的结果,其中:$c_t \in C, r_i \in RA$。显然,$fail$函数的执行过程是线性的,可以在O(n)时间内完成。至此,可以获得函数FR的定义:

$FR = \{r_i \mid r_i \in RA, fail(c_t, r_i)\}$

通过定义函数FR可以知道,如果$FR = \emptyset$,那么软件安装包的PSL策略r_i转换完成。

4 测试与评价

4.1 评价标准

需要考虑多方面的因素来评价智能移动终端安全防护系统的有效性,传统计算机中对攻击的防范能力、对原系统性能的影响对智能移动终端安全防护系统的评价也适用。另外,模型和算法的计算复杂性以及对电池电量消耗的影响也是两个非常重要的因素。充分考虑智能移动终端的特点和MacDroid内核层访问控制技术的优势,从对不同层恶意软件的检测能力、对系统性能的影响和对系统电池电量消耗的影响3个方面来验证MacDroid的有效性。

4.2 测试环境设置

Android系统由于开放性、定制的便利性、用户使用的友好性迅速成为市场占有率最高的智能移动终端操作系统。为了便于比较模型和算法在不同配置硬件上的性能表现,选择两款不同配置的智能手机(samsung i9300和HTC Hero(G3))和一款基于Android的平版电脑(Google Nexus 7)作为实验平台。由于实验需要修改Android各层的源代码,因此选择完全开源并且硬件适配性更好的Android发行版——CyanogenMod作为实验的基础公用软件平台。

4.3 不同层的恶意软件检测测试

为了验证MacDroid对智能移动终端操作系统不同层次的恶意软件的防护效果,在Android CyanogenMod 11(内核采用Linux kernel 3.4.10)和Android CyanogenMod 7.1(内核采用Linux kernel 2.3.5)上分别实现了MacDroid模型。实现工作包括:在kernel中加入一些策略判决的挂载点,将基本判决逻辑实现为Linux可加载内核模块LKM,将策略编译和注入工具实现为本地可执行程序,将文本格式策略和内核可识别策略保存为文件系统文件。

带有MacDroid内核层访问控制模块功能的Android智能移动终端设备运行时,如果某项操作违反了PSL安全策略,系统会弹出报错提示,并打印出内核中的判决信息,如图6所示。

```
type=2000 audit(1262304372.200:12):inode_perm
    (subj: undef, obj: deny, global: allow)
subj=(comm: Binder Thread #, pid: 964, ino: 0, , 0, 0)
    unknow_op
    obj=(ino: 209, video0, major: 81, minor: 0)
      untrusted dev
```

图6 违反MacDroid内核安全策略的终端场景图

(下转第380页)

attribute-based encryption[C]// Proc. of the 2007 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 2007: 321-334.
[4] 苏金树，曹丹，王小峰，等. 属性基加密机制[J]. 软件学报，2011，22(6)：1299-1315.
[5] KAPADIA A, TSANG P P, SMITH S W. Attribute-based publishing with hidden credentials and hidden policies[C]// Proceedings USA of 14th Annual Network & Distributed System Security Symposium (NDSS 2007). San Diego, California, USA: NDSS, 2007: 179-192.
[6] NISHIDE T, YONEYAAMA K, OHTA K. Attribute-based encryption with partially hidden encryptor-specified access structures[C]// 6th International Conference on Applied Cryptography and Network Security. Berlin: Springer Berlin Heidelberg, 2008: 111-129.
[7] LAI J, DENG R H, LI Y. Fully secure cipertext-policy hiding CP-ABE[M]// Information Security Practice and Experience. Berlin Heidelberg: Springer, 2011: 24-39.
[8] 宋衍，韩臻，刘凤梅，等. 基于访问树的策略隐藏属性加密方案[J]. 通信学报，2015，36(9)：119-126.
[9] 解理，任艳丽. 隐藏访问结构的高效基于属性加密方案[J]. 西安电子科技大学学报(自然科学报)，2015，42(3)：97-102.
[10] 雷蕾，蔡权伟，荆继武，等. 支持策略隐藏的加密云存储访问控制机制[J]. 软件学报，2016，27(6)：1432-1450.
[11] XU R, WANG Y, LANG B. A Tree-Based CP-ABE Scheme with Hidden Policy Supporting Secure Data Sharing in Cloud Computing[C]// 2013 International Conference on Advanced Cloud and Big Data (CBD). Piscataway: IEEE, 2013: 51-57.
[12] 杜瑞颖，沈剑，陈晶，等. 基于策略隐藏属性加密的云访问控制方案[J]. 武汉大学学报(理学版)，2016，62(3)：242-248.
[13] 汪海萍，赵晶晶. 隐藏访问结构的密文策略的属性基加密方案[J]. 计算机科学，2016，43(2)：175-178，198.

(上接第 356 页)

在内核层、本地层和应用层分别设计了破坏 PSL 安全策略的恶意软件，内核层恶意软件的表现形式为 Rootkit，本地层恶意软件的表现形式为本地可执行程序，应用层恶意软件为 apk 应用程序。由于智能移动终端的安全性研究是新兴的研究领域，目前尚无对恶意软件检测效能进行评价的权威数据集，因此自行设计了测试数据集，数据集的分布如表 1 所列。

表 1 3 个层次上被测软件的数据集分布

	总被测软件数	恶意软件数	正常软件数
内核层	100	20	80
本地层	100	20	80
应用层	100	20	80

为了评价 MacDroid 模型对 3 个层次上恶意软件的检测和防范效果，使用漏报率(False Negative Rate，FNR)、误报率(False Positive Rate，FPR)和检测率(Detection Rate，DR) 3 个参数来定量评价。3 个参数的计算方法如下：

$$FNR=\frac{\text{漏检恶意软件数量}}{\text{被测软件总数量}}$$

$$FPR=\frac{\text{误报恶意软件数量}}{\text{被测软件总数量}}$$

$$DR=\frac{\text{检测出的恶意软件数量}}{\text{恶意软件总数量}}$$

不同层的恶意软件检测测试属于模型的功能完备性测试，与具体硬件设备的配置和性能无关，因此只在 Google Nexus5 手机上进行测试。测试结果统计如表 2 所列。

表 2 3 个层次上层恶意软件漏报率、误报率和检测率统计表

	漏报数	漏报率/%	误报数	误报率/%	检测数	检测率/%
内核层	4	4	5	5	16	80
本地层	5	5	7	7	15	75
应用层	7	7	10	10	13	65

分析上面的测试结果可知，MacDroid 模型对 3 个不同层次的恶意软件检测的漏报率和误报率都较低，检测率较高。对 3 个层次之间进行比较可以发现，模型对内核层恶意软件的检测效果更好，原因在于：对于越底层的软件，在内核中获得的其主客体和动作行为的特征越准确，与 PSL 安全策略的匹配也越精确，因此其更容易被 PSL 安全策略控制。

结束语 本文主要从智能移动终端内核层安全防护角度深入分析和讨论了终端内核层安全防护的必要性，提出了一种可验证的轻量级内核层访问控制模型，并对模型进行了形式化描述，最后通过实验验证了模型在 3 种典型智能移动终端上的可用性和不足。

参 考 文 献

[1] PIRRETTI M, TRAYNOR P, MCDANIEL P, et al. Secure Atrribute-Based Systems[J]. Journal of Computer Security, 2010, 18(5): 799-837.
[2] ION I, DRAGOVIC B, CRISPO B. Extending the Java Virtual Machine to Enforce Fine-Grained Security Policies in Mobile Devices[C]// Proc. of the Annual Computer Security Applications Conference. 2007: 233-242.
[3] ZHANG X W, ACIIÇMEZ O, SEIFER J. A Trusted Mobile Phone Reference Architecture via Secure Kernel[C]// Proc. of the ACM workshop on Scalable Trusted Computing. 2007: 7-14.
[4] ENCK, WILLIAM ONGTANG, et al. On Lightweight Phone Application Certification[C]// Proceedings of the 16th ACM Conference on Computer and Communications Security. 2009: 235-245.
[5] KIRKPATRICK M S, BERTINO E. Enforcing Spatial Constraints for Mobile RBAC Systems[C]// Proc. of the 15th ACM Symposium on Access Control Models and Technologies. 2010: 99-108.
[6] NAUMAN M, KHAN S, ZHANG X W, et al. Beyond Kernel-Level Integrity Measurement Enabling Remote Attestation for the Android Platform[C]// International Conference on Trust and Trustworthy Computing. 2010: 1-15.
[7] NSA[OL]. http://selinuxproject.org/page/SEAndroid.
[8] 黄琳雅. 基于内核的 Android 文件访问控制研究[D]. 北京：北京邮电大学，2012.
[9] 易筱茂. 面向 Android 操作系统的强制访问控制研究[D]. 北京：中国科学院大学，2015.
[10] 卿斯汉. Android 安全的研究现状与展望[J]. 电信科学，2016(10)：1-8.