移动终端系统的访问控制框架

刘 伟 梁洪亮

(中国科学院软件研究所国家基础软件工程研究中心 北京 100080)

摘 要 随着移动计算的不断普及,针对移动终端的安全问题日益受到重视。传统的访问控制技术能够解决机密性和完整性问题,但没有充分考虑可用性要求。移动终端同个人计算机相比更容易耗尽系统资源,受到拒绝服务攻击。本文提出一种应用于移动终端的访问控制框架,扩充 DTE 机制,保证数据机密性和完整性,并允许关键应用预留动态资源(如 CPU 时间、物理内存页面等),系统对预留的资源进行统一的调度和管理,防止关键应用与其它程序竞争系统资源时产生冲突,保证系统对关键应用的响应,提高系统的可用性。

关键词 移动终端安全,访问控制,资源预留,可用性

Access Control Framework for Mobile Terminal Systems

LIU Wei LIANG Hong-Liang

(National Engineering Research Center of Fundamental Software, Institute of Software, The Chinese Academy of Sciences, Beijing 100080)

Abstract With popularity of mobile computing, security issues of mobile terminals recently became a hot research area. Traditional access control technologies can effectively resolve confidentiality and integrity problems of information, but do not pay enough consideration to availability. Compared with personal computers, mobile terminals are easier under Denial of Service attacks by exhausting system resource. In this article, we provided an access control framework for mobile terminals. Extending DTE mechanism to guarantee confidentiality and integrity of information, the framework allows key applications to reserve dynamic resource (such as CPU time and physical memory pages). Managed and scheduled by mobile terminals, reserved resource can be used to avoid resource conflict between key applications and others, which improve response and availability of mobile terminals.

Keywords Mobile terminal security, Access control, Resource reservation, Availability

1 引言

随着互联网和无线技术的飞速发展,人们对于移动计算的需求不断提高,移动终端(智能手机、个人数字助理等)成为日常生活的重要工具。由于便携性和计算能力不断提高,移动终端现在大量应用于商务场合。与此同时,针对移动终端的安全威胁日益受到人们的重视。安全事件层出不穷:手机病毒和恶意程序泄露、篡改个人信息,占用网络浪费通讯费用,耗尽资源导致系统无法响应,甚至造成系统崩溃,妨碍正常使用。

用户对信息系统的安全需求包括:机密性、完整性、可用性等方面。传统访问控制技术能够有效解决信息的机密性和完整性,但没有充分考虑系统的可用性。传统访问控制技术的研究平台并非针对移动终端。虽然移动终端的硬件配置不断提高,但同服务器和个人电脑相比计算资源非常有限。恶意代码或设计/实现较差的程序更容易耗尽资源,导致系统无法响应关键应用(例如语音通讯、闹钟等),从而达到拒绝服务攻击(DoS)的目的,因此不能将传统访问控制技术直接应用于移动终端。与其它平台相比,移动终端具有以下特性:

- 依赖于公共媒介(无线网络)通讯;
- 很小的体积和很强的移动性导致容易丢失和被盗;
- 有限的计算和电池资源;
- 单用户系统。

移动终端的特性对访问控制技术提出了新要求:可用性、易用性和高性能。可用性是指必须对关键应用及时响应;易用性是指访问控制规则应该尽可能简单,过于复杂的规则将导致用户失去耐心,甚至关闭访问控制机制;高性能是指硬件水平决定了移动终端无法进行大规模的数据计算,数据操作过多使性能严重下降。

2 相关工作

一些组织和研究人员已经关注如何提高移动平台的安全性,并提出了相应的安全结构。可信计算组织(TCG)定义了一个通用的系统结构[1],主要用于个人电脑、个人数字助理和移动电话等平台。在 TCG 系统中,可信平台至少要提供 3 个基本特性:保护能力、完整性衡量和完整性报告。在保护能力特性中,TCG 定义了系统级的访问控制机制,主要保护对敏感数据(例如内存和寄存器等本地存储设备)的访问。可信移动平台(TMP)是针对移动无线平台设计的安全结构,包括硬件结构、软件结构[2]以及协议规范。TMP 的软件结构利用硬件特性(例如可信平台模块和基于硬件的域隔离机制)来提高整个平台的安全性。可信移动设备支持两种访问控制模型:自主模型和系统强制模型。这两种模型都是建立在基于硬件的域隔离机制之上,保证对数据和资源的访问都是经过授权

*)本课题受法国电信 SeLP 项目资金支持,本文工作由中科院软件所与法国电信北京研发中心共同承担。刘 伟 博士研究生;梁洪亮 博士,副研究员。

的。消费电子 Linux 论坛(CELF)在消费电子设备上使用 Linux 开放平台。在定义的安全规范[3]中,CELF 从设备制造商和用户的角度提出消费电子安全性保证,主要包括:数据的安全性、完整性,隐私的保护等。

尽管上述工作都提出了自己的安全结构,但没有充分考虑移动终端的硬件水平,因而无法保证在有限资源的条件下及时响应关键应用。本文提出一种针对移动终端系统的访问控制框架,不但保护信息的机密性和完整性,同时兼顾关键应用的响应,提高系统的可用性。

3 系统控制框架

移动终端是单用户系统,系统中实施访问控制的主体是进程。被保护的客体分为两大类:

1)静态资源:系统中静态存在的资源,包括硬件接口和敏感数据两种。

硬件接口:包括蓝牙、红外和无线网卡等设备接口;

敏感数据:包括保存在 SIM 卡、移动终端和移动存储设备上的个人数据等。

2)动态资源:系统中动态分配的资源,包括 CPU 时间、物理内存页面和电池电量等。

很多经典的文章对静态资源的访问控制机制进行了详细阐述,并给出成熟的访问控制模型。在众多访问控制机制中,DTE 机制[4]由于灵活简单、功能强大的特性受到青睐,并应用于主流操作系统中(例如 SELinux)。我们选择 DTE 机制作为移动终端系统访问控制的基本框架,来控制对静态资源的访问,并根据不同安全需求配置 DTE 机制。在保证信息的机密性和完整性的基础上,我们对访问控制机制进行扩展,着重提高系统的可用性。文[5,6]提出资源预留的概念,操作系统为实时进程预留资源,避免和其它进程由于竞争访问资源而导致实时性能破坏,从而保证系统对实时进程的及时响应。在访问控制框架中加入对动态资源的控制,允许关键应用预留系统资源,从而能够保证系统的及时响应。

在移动终端系统的访问控制框架(如图 1 所示)中,主体向安全服务器提交对客体的访问请求,访问代理截获访问请求,从安全策略库中读取安全策略,根据客体的类型将访问请求及安全策略分别提交给预留监控器(预留动态资源客体)和访问监控器(访问静态资源客体)。预留监控器和访问监控器分别做出访问决策后,将结果返回访问代理,并由访问代理将访问决策通知主体。

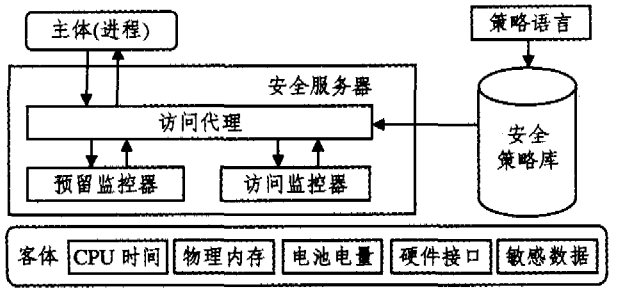

图 1 移动终端系统的访问控制框架

4 功能详细设计

安全服务器主要由 3 部分构成:访问代理、预留监控器和访问监控器。访问代理截获主体的访问请求,分发给预留监控器和访问监控器,并将它们返回的结果通知主体。预留监控器负责裁决预留请求,并进行预留调度和管理。访问监控器根据访问控制模型对主体的访问客体请求做出判断。

4.1 访问代理

主体向安全服务器发出访问请求,由访问代理截获请求,并从系统安全策略库中读取安全策略。根据访问请求类型不同,访问代理将访问请求和对应的安全策略分别转交预留监控器和访问监控器进行裁决,裁决结果返回访问代理,最后由访问代理将结果返回主体。具体流程如下:

1)访问代理截获访问请求;

2) 从系统中读取安全策略;

3) 将安全策略分割为预留策略和访问策略;

4) 根据访问类型,将预留策略和预留请求转交预留监控器,将访问策略和访问请求转交访问监控器;

5) 预留监控器和访问监控器分别做出裁决;

6) 访问代理处理裁决并将结果通知主体。

4.2 访问监控器

在移动终端系统中,访问控制不但要考虑安全性保证,同时要考虑易用性,方便用户配置和管理。访问监控器采用 DTE 策略:主体和域相关联,客体和型相关联。如果一个域不能以某种访问模式访问某个型,则这个域中的主体不能以该访问模式访问属于该型的客体。

4.2.1 型的划分

移动终端系统的客体按照安全需求被分为 4 种型:interface_t、sensitive_t、system_t 和 other_t。硬件接口属于 interface_t 型,个人数据、数字产品等敏感信息属于 sensitive_t 型,系统文件和数据属于 system_t 型,一般客体属于 other_t 型。

4.2.2 域的划分

根据不同的安全功能,主体被分为 trusted_d、certified_d 和 untrusted_d 3 个域。设备制造商预置的程序(例如 phone、notes 等)是移动终端的基本功能和关键应用,属于 trusted_d 域,该域中的主体能够访问移动终端中所有的被保护客体和资源。经过第三方验证的程序属于 certified_d 域,该域中的主体能够访问一般客体以及数字版权产品等敏感数据,即能够访问属于 interface_t、sensitive_t 和 other_t 型的客体。没有经过验证的程序和移动代码都属于 untrusted_d 域,该域中的主体不能访问被保护的客体和敏感信息,只能访问属于 other_t 型的客体。

属于 untrusted_d 域的主体能通过数字签名等验证机制转换到 certified_d 域中,同样,属于 certified_d 域的主体由于验证失败或证书过期而转换到 untrusted_d 域中。属于 certified_d 和 untrusted_d 域的主体不能转换到 trusted_d 域中。

4.3 预留监控器

预留监控器在收到访问代理转交的预留请求后,根据安全策略和系统当前预留资源进行裁决,并负责对系统中所有的预留资源进行调度和管理。

4.3.1 预留类型

· CPU 时间预留

CPU 时间预留是指主体在处理器上占用特定的预留时间。系统为主体在每个周期内预留一定的处理器时间。预留请求包含两个参数:计算时间和周期。计算时间是指在预留周期内的持续时间;周期是指预留应该持续的多少个周期。

· 物理内存预留

物理内存预留代表一系列物理内存页面。物理页面是离散资源，因此支持简单的离散预留。物理页面能够被锁定、换入或换出，预留物理页面的主体能够避免同系统中其它进程在使用物理内存页面上产生冲突。

· 电量预留

CPU、LCD、音频、视频等设备根据不同电量预留采用相应的省电模式[7]。

· 网络带宽预留

网络带宽预留是指在网络设备上预留执行时间，它的单位是特定大小的网络包传输的数目。

· 预留集合

预留集合是指对主体预留若干种类型资源的集中管理，可以包括一种或多种已定义的预留。例如预留集合 Set_i 可以表示为：$Set_i=\{CPU\_P_i, Power\_P_i\}$。

4.3.2 预留管理

预留管理包括3个方面：准入控制根据系统当前的预留情况以及主体请求决定是否允许预留请求；预留调度采取不同的调度模式最大限度地利用系统的资源；预留执行保证主体能够使用系统为其预留的资源，同时对使用超过预留部分的资源进行处理。准入控制、调度策略和预留执行是相互关联的，采用怎样的准入控制，会对调度策略和预留执行产生影响，反之亦然。

由于移动终端的资源有限，多个主体同时提出预留请求并被允许后，预留监控器为主体预留资源是多个预留请求中的最大集合，描述如下：

$system\_reservation = \{\max(CPU\_P_1, \cdots, CPU\_P_i), \max(Power\_P_1, \cdots, Power\_P_j), \max(Power\_P_1, \cdots, Power\_P_k), \max(Network\_P_1, \cdots, Network\_P_m)\}$

4.3.2.1 准入控制

预留管理器对主体提出的预留请求做出裁决，确定哪些预留能够被执行。移动终端要保证具有响应时间限制的主体能够预留资源，这种类型的主体都属于 trusted_d 域。只有属于 trusted_d 域中的主体才能预留资源，其它域中的主体提出的预留请求将被拒绝。

移动终端系统中已经预留的资源也会对新的预留请求产生影响。如果预留的资源过多，会造成其它程序没有足够的资源运行，影响系统的整体性能。如果预留请求的资源超过系统定义的限额(quota)，预留监控器将拒绝该请求。

4.3.2.2 预留执行

预留监控器要确保主体能够使用系统为其预留的资源，但主体可能在用完预留的资源后继续运行。根据主体使用预留资源的类型，预留执行把主体当前使用预留资源的情况通知预留调度，由预留调度根据调度策略决定是否允许主体继续运行。对于非连续的资源(如物理内存页面、网络缓冲区等)，预留执行很容易统计当前使用情况，而对于CPU时间等资源则需要采取不同的措施。以CPU时间预留为例(图2)，在一个预留周期内，主体使用自己预留的资源阶段称为硬预留。这个阶段中，主体只能被比自己优先级高的主体抢占预留，计时器1通知预留调度该阶段结束。如果系统预留的资源没有用尽，主体继续执行，将使用系统为其它主体预留的资源，这个阶段称为软预留。此时主体能被预留资源的主体或者比自己优先级高的主体抢占预留，计时器2通知预留调度该阶段结束。这个预留周期内，系统预留的资源已经用尽，主体继续执行，无法使用预留资源，同普通主体一起调度。

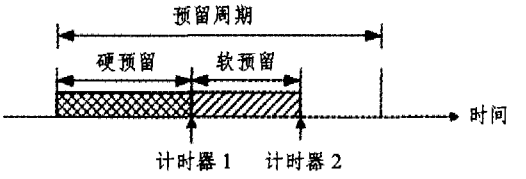

图2 CPU时间预留执行

4.3.2.3 预留调度

通常的资源预留结构中，主体与预留资源绑定，保证主体需要使用预留资源的时候不会与其它进程发生冲突。由于移动终端的资源有限，采用绑定的方式可能导致预留资源过多，影响系统性能。移动终端是单用户系统，在特定的时间内只有一个主体同用户进行交互。例如，用户进行语音通讯的时候，系统不需要对备忘录进程进行响应，在语音通讯结束后响应即可。由于系统中的预留资源由多个主体共享，因此存在多个主体对于预留资源同时进行访问的可能，系统需要对访问预留资源进行调度。预留监控器采用基于优先级的调度策略，高优先级的主体能够抢占正在运行的低优先级主体的预留资源，低优先级主体被置于等待队列中。当高优先级主体运行完成后，重新获得预留资源。

预留调度算法描述如下：

1)主体 p_1 获得系统中的预留资源集，系统中可用的预留资源为：

$available\_set = system\_reservation - p1 \rightarrow reservation\_set$;

2)主体 p_2 向预留监控器请求使用预留资源；

if($p_2 \rightarrow reservation\_set \subset available\_set$)

{ p_2 execution} //能够满足预留要求，p_2 执行

else if($p_2 \rightarrow riority <= p1 \rightarrow priority$)

$add\_to\_queue(p_2)$; // p_2 加入等待队列

3)p_1 和 p_2 在使用预留资源时产生冲突，p_1 在硬预留阶段，只有优先级更高的主体才能抢占预留资源；

if($p_2 \rightarrow priority > p_1 \rightarrow priority$)

$switch\_context(p_1, p_2)$;

4)$Timer_1$ 到期，p_1 在软预留阶段，被等待队列中的 p_2 抢占预留资源；

if($no\_empty\_queue()$){

$p_2 = wake\_up\_queue()$;

$switch\_context(p_1, p_2)$; }

5)$Timer_2$ 到期，当前使用预留资源的进程用尽预留，同普通进程一起调度

$p_1 = current\_reservation()$;

$p_1 \rightarrow priority = 0$; //降低优先级

$schedule()$;

4.4 策略语言

DTE策略使用描述性语言进行定制，用户能够灵活方便地配置和管理。在移动终端系统中，策略描述文件被编译成二进制格式，在系统启动时载入安全策略库，供安全服务器使用。

策略语言主要包括5个部分：域的定义、型的定义、预留的定义、访问矩阵和访问限制。域的定义描述主体和域的对应关系；型的定义描述客体和型的对应关系；预留的定义描述预留的资源集以及预留主体的优先级；访问控制矩阵定义域

对型的访问权限；访问限制描述在整个访问控制框架中用到的限制关系，例如对访问时间的限制。

```
#Domain Definitions                  #Access Control Materix
/bin/phone=trusted_d                 (trusted_d,system_t,all)
/usr/bin/mplayer=certified_d         (certified_d,sensitive_t,rw)
#Type Definitions                    #Constraints
/ect/network=system_t                (certified_d,sensitivie_t,
/usr/local/abc. mpe=sensitive_t      Timeinterval! 9:00! 22:00)

#Reservation Definitions
CPU_P1=20ms,*
Memory_P1=30
Power_P1=strict
Set1={CPU_P1,Memory_P1,}
Set2={Power_P1}
/bin/phone 90
```

图 3　策略语言示例

进一步的工作和结论　在预留执行中，系统对主体使用超过预留资源的情况进行了处理。另一方面，用户对主体运行所需要的资源无法准确判断，定义的预留资源可能大于实际用到的资源，造成系统资源浪费，降低系统性能。对于这种情况，需要进一步改进资源预留模型，以便用户可以根据资源的实际使用情况进行调整。另外，在系统运行过程中，一些非周期性进程需要动态修改其预留资源，需要在模型定义以及框架方面做进一步的改进。

本文设计了一种应用于移动终端系统的访问控制框架，不但能够保证数据的机密性和完整性要求，而且满足系统对关键应用及时响应的要求，提高移动终端的可用性。在该访问控制框架中，移动终端客体被分为两类：静态客体和动态客体。主体提出的访问请求通过访问代理转交给访问监控器和预留监控器，前者负责对静态客体的访问进行控制，后者负责对主体提出的动态客体预留请求进行裁决并对系统预留资源进行管理。通过预留资源的实施，系统避免了关键应用与其它程序之间产生的资源使用冲突。在访问控制框架中，我们使用一种扩充 DTE 策略的描述性语言来定义访问控制策略，方便用户的配置和管理。

参 考 文 献

1　Trusted Computing Group. TCG Specification Architecture Overview. V1. 2, Apr. 2004. https://www. trustedcomputinggroup. org/

2　Trusted Mobile Platform. Software Architecture Description, 2004. http://www. trusted-mobile. org/

3　Consumer Electronics Linux Forum. CELF Specification v1. 0, Jun 2004. http://www. celinuxforum. org/

4　Badger L, Sterne D F, Sherman D L, et al. Practical Domain and Type Enforcement for UNIX. In: IEEE Symposium on Security and Privacy, Oakland, California, May 1995. 66～77

5　Rajkumar R, Juvva K, Molano A, et al. Resource Kernels: A Resource-Centric Approach to Real-Time and Multimedia Systems. In: Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking, January 1998

6　Mercer C W. Operating System Resource Reservation for Real-Time and Multimedia Applications. Carnegie Mellon University, 1997

7　Scordino C, Lipari G. Using Resource Reservation Techniques for Power-Aware Scheduling. In: Proc. of the 4th ACM Intl. Conf. on Embedded Software, Pisa, Italy, Sept. 2004

（上接第 78 页）

状态变迁规则库：用于存放事件和事件状态变迁规则。每个状态变迁规则包括事件类型、起始状态、变迁状态和变迁事件，一种事件可能具有多条状态变迁规则。

状态树管理模块：该模块主要根据事件匹配模块匹配的结果对状态树进行管理，该模块对每一个事务状态机维持一个 TSM 节点，每个 TSM 节点下有一个对应的状态变迁树，每次事件到达时，以事件的状态机标识信息为索引找到相应的状态变迁树。然后，根据变迁规则，通过推理，使状态树向每一个可能的方向进行生长。当状态变迁树不能生长时，通知异常处理模块进行异常判断，并对已无生长可能的分支进行剪枝处理。

异常处理模块：对包处理模块、状态树管理模块中产生的异常信息进行格式化处理，并且能对出错原因进行一定的推理分析，然后将分析结果发送给数据存储模块。

数据存储模块：对收到的报文信息、正在维护的 TSM 树信息以及异常信息进行存储管理，以便用户查看。

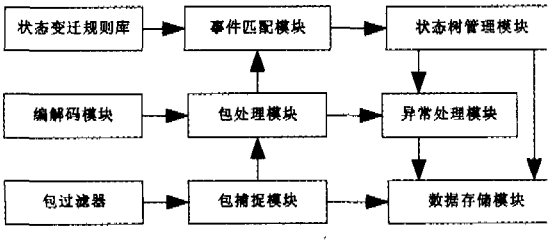

图 3　软件结构图

总结　通信协议一致性测试是一个十分活跃的研究领域，在线测试作为一种协议一致性测试方法具有其独特的优势，由于这种测试本身不对被测系统的正常运行带来干扰，因此可以实现对被测系统长时间的测试。本文对在线测试在 BACnet 通信协议的应用进行了讨论，给出了基于有限状态机模型的 BACnet 协议状态机测试方法和软件框架。

参 考 文 献

1　吴建平，尹霞. 基于形式化方法的协议测试理论[J]. 清华大学学报(自然科学版)，2001，41(4/5)

2　ASHRAE. BACnet: A Data Communication Protocol for Building Automation and Control Networks 135-2001

3　Spitsyna N, Trenkaev V. FSM Based Interoperability Testing of Communication Protocols. Control and Communications, the 2003 IEEE-Siberian Conference on, 2003. 20 ～ 23

4　Zhao Yixin , Yin Xia , Han Bo, et al. Online Test System Applied in Routing Protocol Test Modeling. Analysis and Simulation of Computer and Telecommunication Systems. In: Proceedings of Ninth International Symposium on, Aug. 2001. 331～338

5　赵邑新，吴建平. 应用于在线测试的状态判定算法. 电子学报，2000，28(11)：83～87