doi:10.11918/j.issn.0367-6234.2016.05.012

进程安全保护技术综述

赵剑锋, 史 岗, 孟 丹

(中国科学院 信息工程研究所, 北京 100093)

摘 要: 进程安全保护是信息安全领域的一个研究热点,可以应用于计算机安全系统、网站保护、杀毒软件自保护、数字产权保护等方面. 在信息安全研究的背景下,总结了进程所受到的威胁,包括软件和硬件方面的威胁;结合实际案例对进程安全保护技术从硬件和软件方面进行了详细分类,分析了每种技术的特点和不足;指出现在研究存在的问题,并展望了未来关于进程安全保护的研究方向和重点.

关键词: 进程; 安全保护;分类;研究方向

中图分类号: TP309.2 **文献标志码**: A **文章编号**: 0367-6234(2016)05-0077-08

A survey on the studies of security guard technology for process

ZHAO Jianfeng, SHI Gang, MENG Dan

(Institute of Information Engineering, Chinese Academy of Sciences, 100093 Beijing, China)

Abstract: Security and protection for process is a hot point in the field of information security, which can be applied to computer security system, anti-virus software, the protection of digital property rights, etc. In the context of information security, this paper summarizes the threats of the process including software threat and hardware threat, and combines the actual case. It classifies the security protection technology for process from hardware and software aspects, as well as analyzes the characteristics of each technology and its shortcomings. Finally the paper concludes the problems in the current research, and looks forward to the future research direction and focus on security protection technology for process.

Keywords: process; security protection; classification; research direction

信息安全主要涵盖以下方面的内容,也就是需保证信息的几个特性:机密性、真实性、完整性、未授权拷贝和信息所寄生系统的安全性. 信息安全概念本身包括的外延范围很大,其中包括防范商业企业机密泄露、防范未成年人对不良信息的浏览、防范个人信息的泄露等. 信息安全是指信息系统(包括计算机硬件、计算机软件、数据、人员、物理环境及其基础设施)受到保护,不受偶然的或人为恶意的原因而遭到破坏、更改、泄露,保证系统连续可靠地运行,保证信息服务不中断,最终实现业务连续性[1-10].

为了抵御花样不断翻新的安全威胁,计算机安全界相关研究人员进行了广泛深入的研究,制定了一系列安全相关标准[11-16],同时也开发了各种各样的安全产品. 研究发现,位于计算机系统底层的基础系统的安全在很大程度上决定了整个系统的安全,同时位于上层的应用软件的安全也占有一定地位. 要想构建安全的计算机系统,就应当加强以操作系统为代表的软件的安全性. 基础软件和应用软件都是以程序的形式存在的. 程序是静态的,程序要想起作用,就要转变成动态的进程. 因此,进程安全保护技术是信息安全的重要研究热点之一. 进程是一个具有一定独立功能的程序基于某个数据集合的一次运行活动,它是操作系统动态执行、调度的基本单元. 每个进程有自己的地址空间,一般包括代码区域、数据区域和堆栈. 代码区域存储处理执行的代码;数据区域存储程序变量和进程执行期间使用的动态分配内存;堆栈区域存储着进程活动过程调用的指令和本地变量. 进程是一个动态的“执行中的程序”. 因此,进程的安全就是要保护进程地址空间(代码区域、数据区域和堆栈)中数据的机密性

收稿日期: 2016-01-13.

基金项目: 国家"核高基"科技重大专项基金(2013ZX01029003-001); 国家高技术研究发展计划基金(2012AA01A401).

作者简介: 赵剑锋(1980—),男,博士研究生;
孟 丹(1965—),男,研究员,博士生导师.

通信作者: 赵剑锋, zhaojianfeng@iie.ac.cn.

和完整性. 进程安全与软件安全不同. 软件安全主要保护软件中的智力成果、知识产权不被非法使用,诸如篡改及盗用等. 软件安全的研究主要包括如何防止软件盗版、逆向工程、授权加密以及非法篡改等. 进程安全与计算机安全也不同. 计算机安全所涵盖的范围更广,主要包括:计算机病毒防范与检测、非法访问控制、计算机电磁辐射防范、硬件损坏等. 进程安全只是计算机安全研究中的一个子项.

本文给出了进程受到的威胁和进程安全保护技术的分类,指出进程安全保护技术现在研究存在的问题,并对未来的研究方向进行了讨论.

1　进程受到的威胁

在程序及信息系统的开发过程中,由于项目过于庞大复杂,技术难度相对较高,以及开发时间相对紧迫等一系列困难,一般来说,在所开发出的软件产品中,存在安全性隐患的概率是很大的. 如果这些软件程序在计算机系统上运行,程序就转化为进程. 这时,很多别有用心的人或黑客等就会利用开发过程中的失误或漏洞,对运行的进程进行攻击.

程序在运行时受到的攻击可以分成两类:一种是软件攻击,如溢出攻击、软件病毒;另一种是物理攻击,如通过功耗差分分析方法取得加密密钥和通过电磁手段获取内存中正在运行的代码等.

软件攻击有以下几种:

1)缓冲区溢出. 缓冲区溢出的攻击原理比较简单,但是它比较危险,因为它可以获得系统的最高控制权,此外它很难被检测出来. 缓冲区溢出攻击是一种常见的系统攻击手段,通过在程序的缓冲区写超出其长度的内容,造成缓冲区的溢出,从而破坏程序的堆栈,迫使程序转而执行其他指令,从而达到攻击的目的. 如图1所示,溢出攻击通过修改进程堆栈中的数据,使其转向攻击者设定的特殊地址,达到了破坏进程的目的.

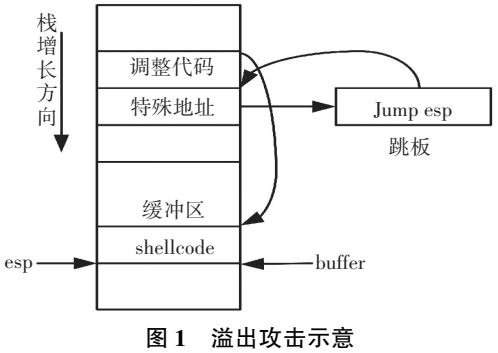

图1　溢出攻击示意

2)病毒. 计算机病毒的本质是程序. 一般它能把自己附加在其他具有正常功能的程序之中,并不断地进行自我复制,然后再去感染其他程序. 病毒能由被感染的程序和系统传播出去,如图2所示,图2(a)表示正常的进程,图2(b)、(c)、(d)表示病毒附加在正常进程的不同位置. 当病毒附加在正在执行的进程中时,无论在进程的头部、尾部还是其他部分,都会导致进程无法完成正常的功能,病毒从而破坏了进程的代码区域、数据区域或堆栈,以此破坏了进程地址空间数据的机密性和完整性.

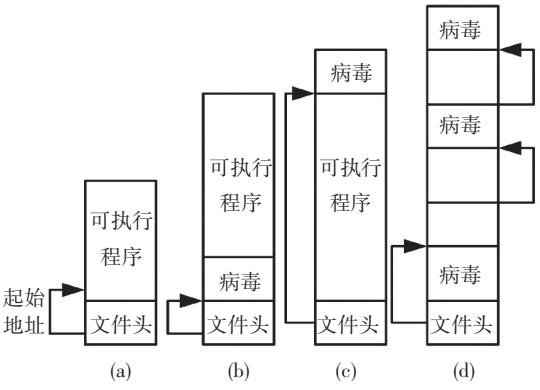

图2　病毒攻击示意

硬件攻击有以下几种:

1)冷启动攻击. 普林斯顿大学前沿基金会和温瑞尔系统公司的研究人员在2008年初,联合发表了一篇题为《鲜为人知的秘密:对密钥的冷启动攻击》的文章,这篇文章从实现细节上介绍了从运行系统获取内存信息的一种新型攻击方式. 一般认为,内存属于数据易失性设备,如果机器一旦断电,内存中的数据就会立刻丢失. 然而,事实并非如此,实际上数据从内存丢失需要一定的时间. 从断开电源开始计算,一到两秒内,内存大部分数据保存如初,并没有什么改变. 如果采用冷却内存的办法,则可以延长数据保持时间,甚至可以到几分钟. 研究人员开发了一系列工具,用来从即将关闭电源的机器内存中提取重要的信息,然后再次启动系统.

2)火线接口攻击. 电气和电子工程师协会的1394接口,一开始是用于取代SCSI的一个高速通信接口. 火线的主要部分是它的高速数据传输速率,这也是它大量应用于音视频传输的原因. 火线接口高速率的一个主要特征是,它可以完全绕过CPU,通过直接内存访问DMA的方式访问内存. 这正是这个接口脆弱、易受攻击的地方.

火线接口具有直接访问内存的能力,这使得能够接触目标机器的渗透测试人员有了攻击机会,他们可以通过重写包含访问控制功能的内存部分,绕过操作系统的密码保护机制. DMA同样可以让渗透测试人员下载小于4GB内存镜像. 即使火线接口仅能访问少于4GB的内存且能使用反病毒软件保护DMA,这类型的访问还是揭示了IEEE1394标准中

重大的安全缺陷. 由于火线接口的热插拔功能，即使目标机器在锁定情况下，攻击同样能够进行.

3）边信道攻击（side channel attack 简称 SCA），又称为侧信道攻击. 针对加密电子设备在运行过程中的时间消耗、功率消耗或是电磁辐射之类的侧信道信息泄露而对加密设备进行攻击的方法被称为边信道攻击. 这类新型攻击的有效性远高于密码分析的数学方法，因此给密码设备带来了严重的威胁. 主要分为电磁攻击和能量攻击两种方式.

常见的功耗分析方法有简单功耗分析（simple power analysis，SPA）、差分功耗分析（differential power analysis，DPA）、高阶差分功耗分析（high order differential power analysis，HODPA）. 简单功耗分析方法利用功耗和操作之间的依赖性，这种方法具有最弱的攻击强度和最小的实施难度[17-19]. 差分功耗分析方法主要利用统计学原理，通过分析功耗和数据之间的相关性来实现攻击，该方法攻击强度较强，应用空间最广[17-18,20-21]. 高阶差分功耗分析方法推广了差分功耗分析方法的应用，它主要用于攻击采用了掩码或盲化技术的密码芯片[22-24].

4）总线攻击. 攻击者通过使用特殊硬件设备搭载到系统总线，利用此硬件设备来窃听总线信息，甚至可以篡改和伪造总线信息，使程序运行得到错误的处理结果. 主要分为三种：一是欺骗攻击，攻击者通过使用伪造的数据块替换掉内存中的数据块来实现；二是重组、拼接攻击，攻击者将内存中不同地址的数据块替换当前块，这种攻击方式可以认为是内存块的空间置换；三是重放、重演攻击，攻击者将一个给定地址内存块的数据块换成历史数据，这种攻击方式可以认为是内存块的时间置换.

2　进程安全保护技术分类

进程安全保护技术可以分成软件和硬件两类. 软件分类主要有如下几种：

1）HOOKAPI 防杀技术. 该技术在 Windows 操作系统信息安全中得到广泛应用，它主要通过截获对某个 API 函数的调用，以此来让程序调用伪 API 函数，Windows 通过调用 TerminateProcessp 这个函数终止进程. 如果对这个函数进行筛选过滤，对需要保护的进程返回一个特殊的值，那么就可以达到保护进程的目的，如图 3 所示.

2）进程隐藏. 想办法把进程隐藏起来，通常是把进程从进程活动链上删除，当从进程活动链上删除之后，受保护的进程就不会被查询到，从而实现了“隐身”操作.

3）三线程保护. 主要运用代码注入技术，目的是让本地线程和远程线程相互监视，只要有其中一方被关闭，另一方马上将其启动. 这种方法发展较为成熟，相对来说采用较多，保护效果也较为不错，除了重启以外，找不到任何方法可以关掉受保护进程.

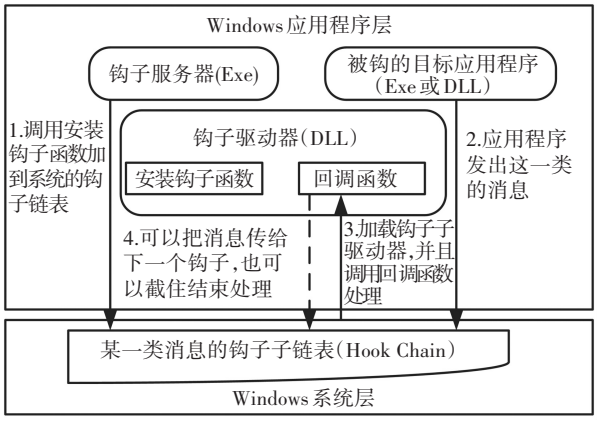

图 3　HOOK API 结构示意

4）动态链接库钩挂. 它是在三线程保护技术基础上所作的改进，主要是让系统进程自动去加载伪动态链接库以及库中的功能.

5）沙箱隔离. 所谓沙箱，其实是一种运行环境，在这种环境中对程序的运行行为制定了安全策略，即什么能做，什么不能做[25].

第一个沙箱系统是 Janus[26]，它采用拦截系统调用的方法来监控程序行为. 它成功地在 Solaris 操作系统上实现了对不可信软件的运行行为的控制. Janus 最新版本[27]利用类似于 Systrace[28]的技术重新实现了系统调用的拦截机制. Chakravyuha[29]与 Janus 系统一样，也是基于内核系统调用拦截技术，成功实现了 Linux 操作系统下的沙箱机制. BlueBox[30]和 Cerb[31]也是这样两种系统.

以上的沙箱系统都存在安全策略难以管理的问题，而 MAPbox[32]提高了安全策略的可配置性与易用性，解决了上述问题. Peterson 和其他研究人员实现了一个通用的系统调用拦截系统[33]，并提供了一组支持二次开发的调用函数. Vassilis Prevelakis 等人在 Linux 和 BSD 操作系统上提出并实现了通过跟踪程序的文件访问操作创建安全策略的沙箱机制[34]. Strata[35]实现了一种沙箱机制，其基础是软件动态转换（Software Dynamic Translation）的运行时程序代码修改技术，其体系结构如图 4、5 所示.

以上保护技术，虽然对进程有一定的保护能力，但是没有保证信息安全的三个特性，即保密性（保证机密信息不被窃听，或窃听者不能了解信息的真实含义）、完整性（保证数据的一致性，防止数据被非法用户篡改）、不可抵赖性（防止用户否认其行

为). 另外,沙箱技术都面临一个可用性的问题,需要手工配置系统的安全策略;无法找到一种便捷有效的方式生成安全策略,难以判断哪些程序行为不会危害系统安全;程序的安全性和完整性难以兼顾.

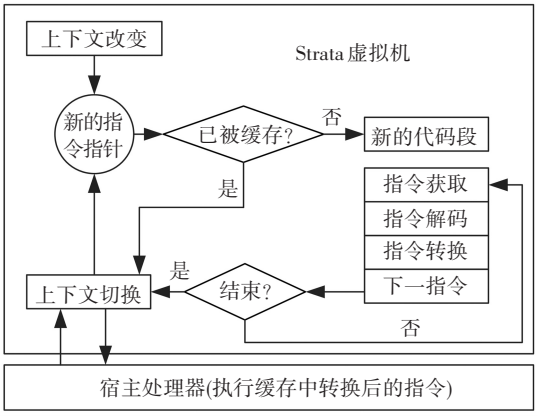

图 4 Strata 指令转换引擎

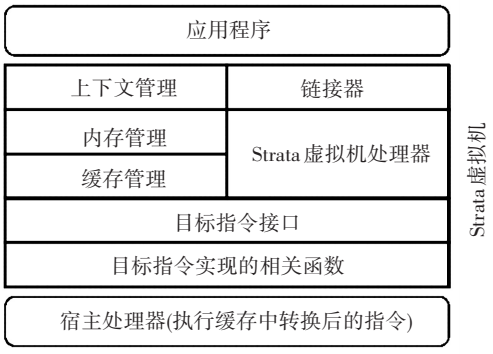

图 5 Strata 体系结构

因此,还需要有相应的硬件技术来对进程进行进一步的保护. 进程的硬件保护主要有以下几种典型的安全体系结构:

1) XOM. XOM(eXecute only memory)[36] 是斯坦福大学计算机系统实验室提出的用于解决数字版权保护问题的系统保护框架,如图 6 所示.

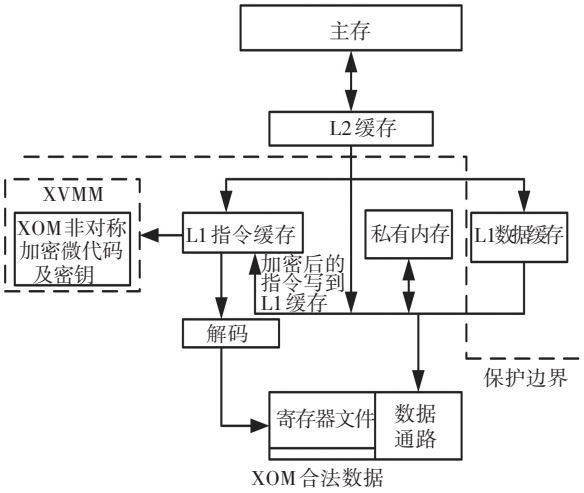

图 6 XOM 架构示意

这种架构的核心是:在 CPU 片内增加加密引擎,不同的程序运行在不同的存储隔间(Compartment)中,存储隔间相互独立,不重叠;同样地,对于数据也是如此. 当系统运行时,程序中能访问属于自己隔间的数据,而不能访问其他隔间的数据. 在 XOM 上运行的程序,事先由软件商进行了加密,所以此程序只能运行在特定的处理器上. 考虑到安全性并且为了提高系统性能,同时采用对称加密算法(处理速度相对快)和非对称加密算法(处理速度相对慢),XOM 使用了密钥共享协议. 首先,XOM 处理器芯片中存储一对非对称密钥对,称之为(K_{xom},K_p),其中,K_{xom} 为私钥,K_p 为公钥. 应用程序发布者使用自己的对称密钥 K_s 对程序进程加密生成相应的密文,然后通过用户处理器公钥 K_p 来加密密钥 K_s,最后将程序密文和加密后的 K_s 一起发给用户. 当系统运行此应用软件时,先使用自己的私钥 K_{xom} 来解密得到软件密钥 K_s,然后通过 K_s 来解密程序并运行. 通过这种方法,软件发布者可以通过使用相应的处理器公钥 K_p 来加密 K_s,从而使得相应的程序只能运行在特定的处理器上,以此达到进程运行时的机密性和完整性保护目的.

2) AEGIS. AEGIS(architecture for tamper-evident and tamper-resistant processing)[37] 系统保护框架是由麻省理工学院的计算机科学实验室提出的. 它的安全前提是:假定处理器和操作系统的部分内核是安全的(security kernel). 它有 TE(tamper-evident)和 PTR(private and authenticated tamper-resistant)两种工作模式. TE 只提供完整性保护,能够探测到所有对数据的篡改行为;PTR 不仅提供完整性保护,同时也提供机密性保护. 在 PTR 模式下,所有片外的数据都是加密的,攻击者不仅无法获取任何具体信息,而且对这些数据进行篡改的行为都可以被系统检测到,如图 7 所示.

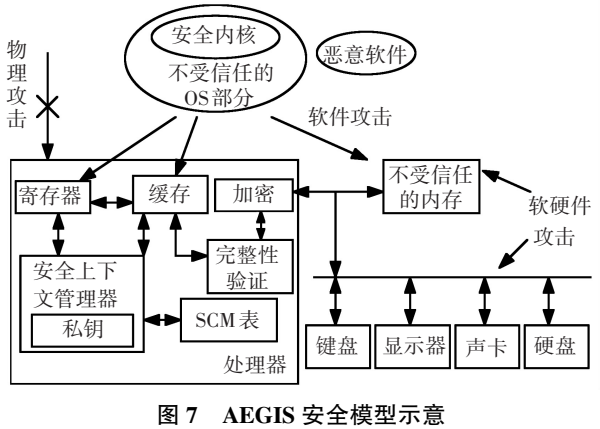

图 7 AEGIS 安全模型示意

在 Security Kernel 中,AEGIS 采用安全上下文管理器 SCM(Secure Context Manager)来维护每一个

进程的相关安全信息. 每条记录有 SPID，H(Prog)，Regs，Hmem，0/1，和(Kstatic，Kdynamic)五个字段组成. 其中：SPID 为安全进程 ID，其值为 0 时表示不受安全保护的普通进程；H(Prog)表示相应的哈希值；Regs 代表相应的寄存器及其值；Hmem 用于完整性校验；0/1 表示该进程的工作保护模式状态(TE 或 PTR)；Kstatic 为对称密钥，用于加解密应用程序，每个应用程序都有一个唯一的 Kstatic，在程序运行过程中保持不变；程序运行过程中产生的数据使用 Kdynamic 进行加密，不同会话产生的 Kdynamic 是不同的，在会话结束时该密钥也就失效了.

AEGIS 架构不仅可以用于软件版权保护，也可以用于认证执行和数字版权管理，这是其优点所在. 但就机密性和完整性而言，AEGIS 采用了直接块加密和 Hash 树校验方法，系统延迟开销较大，效率较低，实用性不是很强，这也正是这种架构的缺点.

3) Ceriums. Cerium[38] 是麻省理工学院提出的另外一种可信处理器，它结合了 XOM 和 AEGIS 处理器优点，通过加密被保护进程的地址空间来实现类似 IBM4758 中的认证执行. 它采用软件方式来加密保护进程，把一个可信微内核放入处理器内部，对被保护进程地址空间的所有操作都会触发这段微内核代码，由它来处理加密地址空间. Ceriums 系统结构如图 8 所示. 另外，在商用 CPU 中，Intel 公司和 AMD 公司也分别推出了自己的增强型安全处理技术，如 Intel 的 TXT[39] 和 AMD 的 SVM[40]. TXT 和 SVM 不提供加密进程空间的功能，它在处理器中增加了新的指令作为动态可信度量根，这条指令可以实现进程的保护执行.

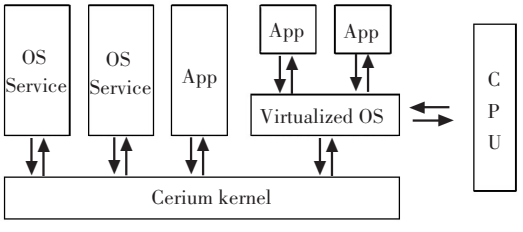

图 8　Cerium 系统结构示意

4) 硬件虚拟化技术. 硬件虚拟化技术[41-44] 通过在操作系统和底层硬件之间插入一层虚拟机监控层 VMM(virtual machine monitor)，与应用程序层的虚拟化技术相比而言，它的权限更高，效率更好，并且有硬件支持.

虚拟机是对真实计算环境的抽象和模拟，所以 VMM 要为每个虚拟机分配一套数据结构，以此来管理各个虚拟机的状态，诸如虚拟处理器的全套寄存器，物理内存的使用情况，虚拟设备的状态等. VMM 调度虚拟机时，将其部分状态恢复到主机系统中，但是并非所有的状态都需要恢复，例如主机 CR3 寄存器中存放的是 VMM 设置的页表物理地址，而不是 Guest OS 设置的值. 主机处理器直接运行 Guest OS 的机器指令，由于 Guest OS 运行在低特权级别，当访问主机系统的特权状态(如写 GDT 寄存器)时，权限不足导致主机处理器产生异常，将运行权自动交还给 VMM. 此外，外部中断的到来也会导致 VMM 的运行. VMM 可以通过模拟操作系统来执行进程调用，并且可以在启动进程以前插入自己的代码，以保护整个进程.

基于虚拟化安全监视的技术可以分为内部监控和外部监控两类. 内部监控在虚拟机中加载内核模块，通过内核模块来拦截目标虚拟机的内部事件，而内核模块本身的安全通过虚拟机管理器来进行保护. Lares[45] 内部监控的架构如图 9 所示. 需要被监控的系统放在目标虚拟机中运行，安全工具放一个与目标虚拟机隔离的安全虚拟域中. 内存保护模块可以根据钩子函数所在的内存页面对进程进行保护，防止攻击者的恶意篡改. SIM[46] 架构也是这个原理，如图 10 所示.

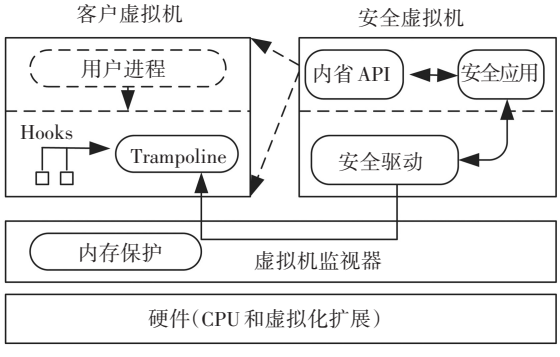

图 9　Lares 内部监控架构

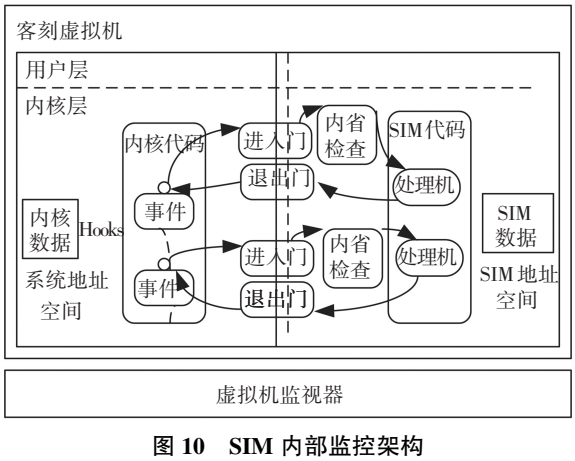

图 10　SIM 内部监控架构

外部监控与内部监控不同，它是在虚拟机管理器中对虚拟机事件进行拦截，在其外部进行检测，如图 11 所示的 Livewire[47] 结构. 外部监控是指在目标虚拟机外部，由位于安全域的安全工具按照某种安

全策略对其进行检测.

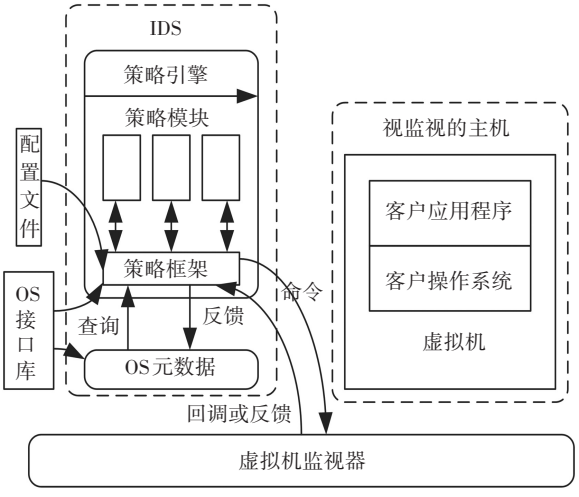

图 11　Livewire 外部监控架构

5) TrustZone 技术. TrustZone 是 ARM 针对消费电子设备安全提出的一种安全系统架构[48-60],目标是防范设备可能遭受到的多种特定威胁,不是所有的威胁. 系统安全是通过将 SoC(system on chip)资源(硬件和软件)划分成两个世界来获得的,即安全子系统对应的安全世界和其他子系统对应的普通世界. AMBA3 AXI 总线系统能确保安全世界的资源不会被普通世界所访问.

就处理器架构而言,每个物理的处理器核提供两个虚拟核:一个非安全核 NS(non-secure)和一个安全核(secure). 非安全核与安全核之间的切换机制叫做 monitor 模式. 非安全核只能访问普通世界的系统资源,而安全核能访问所有资源. 普通世界的软件想进入到 monitor 模式,可以使用 SMC 指令或通过硬件异常机制的一个子集实现. 可以配置 IRQ, FIQ,外部 data abort,外部 prefetch abort 这几个异常进入到 monitor 模式. 图 12 展示了这种切换方式.

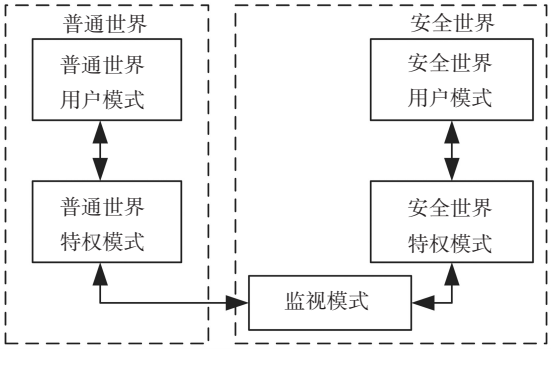

图 12　TrustZone 切换示意

一般情况下,如果普通世界的用户模式需要获取安全世界的服务,它要先进入到普通世界的特权模式,在该模式下调用 SMC,那么处理器进入到 monitor 模式,monitor 模式备份保存普通世界的上下文,随后进入到安全世界的特权模式,此时运行环境变成了安全世界的执行环境,然后再进入到安全世界的用户模式,执行相应的安全服务. 这里把安全世界的用户模式和特权模式分离,是因为通过特权模式中的执行环境是系统级别的,而用户模式的安全服务是应用级别的,两者的提供者通常是不同的. 图 13 是软件架构的展示,安全世界的执行环境要管理用户模式的服务和应用,并给它们提供编程接口.

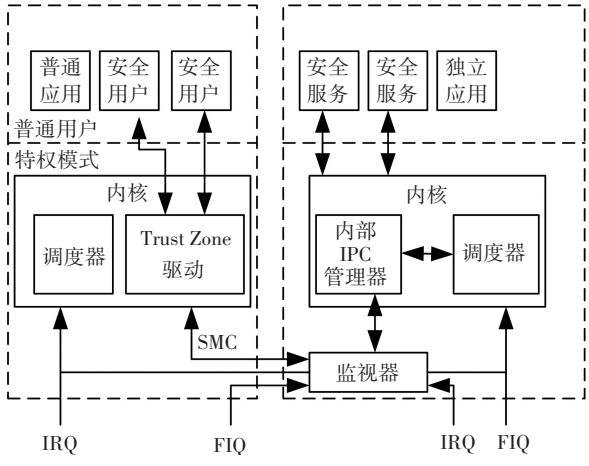

图 13　TrustZone 软件架构示意

3　总结与展望

当前进程安全保护技术主要分成软件和硬件两大类. 软件类中的进程安全保护技术,主要优点是可移植性强,不需要硬件开销,但是存在着不能对信息的机密性、完整性很好保护的缺点. 硬件类中的进程安全保护技术,主要优点是可以做到对进程的机密性、完整性保护,但存在开销大,执行效率低,在实际应用中有局限性的弱点.

因此,下一步的研究应该是,如何结合进程安全保护两类技术的优点,找到好的切入点,以期更好地对进程进行安全保护.

参考文献

[1] McAfee Corp. New paradigm shift: comprehensive security beyond the operating system [R]. California: McAfee, 2012.

[2] MILENA M, ALEKSANDAR M, EMIL J. Using instruction block signatures to counter code injection attacks [J]. ACM Sigarch Computer Architecture News, 2005, 33(1): 1-5.

[3] 冯登国,张阳,张玉清. 信息安全风险评估综述 [J]. 通信学报, 2004, 25 (7): 11-17.

[4] JOHANNES W. Trusted computing building blocks for embedded Linux-based ARM trustzone platforms [C]// Proceedings of the 3rd ACM workshop on Scalable trusted

computing. ACM Press, 2012:21-30.
[5] 邢栩嘉,林闯,蒋屹新. 计算机系统脆弱性评估研究[J]. 计算机学报, 2004, 1(1) : 2-9.
[6] 洪帆,崔国华,付小青. 信息安全概论[M]. 武汉:华中科技大学出版社, 2005: 2-20.
[7] IBM, MICROSOFT. Workshop on advancing computer architecture research (acar-ii): laying a new foundation for IT: computer architecture for 2025 and beyond [R]. IBM, MICROSOFT, 2010.
[8] 金海. 浅析云计算安全的核心问题[J]. 中国计算机学会通讯, 2009, 5(2): 47.
[9] 孙建华. 大规模网络中的入侵检测与病毒防御模型研究[D]. 武汉:华中科技大学, 2005.
[10] 连一峰. 入侵检测综述[J]. 网络安全技术与应用, 2003, 1(1): 46 - 47.
[11] BLAKLEY G R. Twenty years of cryptography in the open literature [C]//IEEE Symposium on Security and Privacy. Oakland: IEEE Press, 1999:106-107.
[12] VIRGIL G. 20 Years of operating system security (UNIX as one focus) [C]//IEEE Symposium on Security and Privacy. Oakland: IEEE Press, 1999:108-110.
[13] STEVE L. Twenty years of evaluation criteria and commercial technology [C]//IEEE Symposium on Security and Privacy. Oakland: IEEE Press, 1999:111-112.
[14] MILLEN J. 20 Years of covert channel modeling and analysis [C]//IEEE Symposium on Security and Privacy. Oakland: IEEE Press, 1999:113-114.
[15] JOHN M. Twenty years of formal methods [C]//IEEE Symposium on Security and Privacy. Oakland: IEEE Press, 1999:115-116.
[16] STEPHEN K. 20 years of network security [C]//IEEE Symposium on Security and Privacy. Oakland: IEEE Press, 1999:117-118.
[17] KOCHER P C. Timing attacks on implementations of Deffie-Hellman, RSA, DSS, and other systems [C]// Advances in Cryptology-CRYPTO'96. California: Springer Press, 1996: 104-113.
[18] KOCHER P, JAFFE J , JUN B. Differential power analysis [C]//Advances in Cryptology-CRYPTO'99. California: Springer Press, 1999: 388-387.
[19] OSWALD E. Enhancing simple power-analysis attacks on elliptic curve cryptosystems[C]//Cryptographic Hardware and Embedded Systems-CHES 2002.Redwood City:Springer Press, 2003:82-97.
[20] BRIER E, CLAVIER C, OLIVIER F. Correlation power analysis with a leakage model [C]//Cryptographic Hardware and Embedded Systems-CHES 2004. Boston: Springer Press, 2004: 16-29.
[21] LI Huiyun, WU Keke, YU Fengqi. Enhanced correlation power analysis attack against trusted systems[J]. Security and Communication Networks, 2011, 4(1): 3-10.
[22] WADDLE J, WAGNER D. Towards efficient second-order power analysis [C]//Cryptographic Hardware and Embedded Systems-CHES 2004. Boston: Springer Press, 2004: 1-15.
[23] NASSAR M, SOUISSI Y, GUILLEY S, et al. RSM: a small and fast countermeasure for AES, secure against 1st and 2nd-order Zero-Offset SCAs[C]//Design, Automation & Test in Europe Conference & Exhibition(DATE 2012). Dresden:IEEE Press, 2012: 1173-1178.
[24] GENELLE L, PROUFF E, QUISQUATER M. Thwarting higher-order side channel analysis with additive and multiplicative maskings[C]//Cryptographic Hardware and Embedded Systems-CHES 2011. Nara: Springer Press, 2011: 240-255.
[25] WANG Sun, ZHANG Liang, SEKAR R, et al. One-way isolation: an effective approach for realizing safe execution environments[C]//Proceedings of Network and Distributed Systems Security Symposium (NDSS'05).San Diego:IEEE Press, 2005: 1-18.
[26] GOLDBERG I, WAGNER D, THOMAS R, et al. A secure environment for untrusted helper applications: confining the wily hacker[C]//Proceedings of the 6th USENIX Security Symposium.San Jose:IEEE Press, 1996: 1-13.
[27] GARFINKEL T. Traps and pitfalls: practical problems in system call interposition based security tools [C]// Proceedings of the ISOC Symposium on Network and Distributed System Security (NDSS'03). San Diego:IEEE Press, 2003: 163-176.
[28] PROVOS N. Improving host security with system call policies[C]//Proceedings of the 12th USENIX Security Symposium. Washington DC: IEEE Press, 2003: 257 - 271.
[29] DAN A, MOHINDRA A, RAMASWAMI R, et al. Chakra Vyuha(CV): a sandbox operating system environment for controlled execution of alien code[R]. Technical Report 20742, IBM T.J. Watson Research Center,1997.
[30] SURESH N C, PAU C. BlueBox: a policy-driven, host-based intrusion detection system [J]. ACM Transactions on Information and System Security, 2003, 6(2): 173-200.
[31] DAWIDEK P J. Cerb: system firewall mechanism [C]// Proceedings of the 12th Usenix Security Symposium. Washington D C: IEEE Press, 2003.
[32] ACHARYA A, RAJE M. Mapbox: using parameterized behavior classes to confine applications[C]//Proceedings of the 9th USENIX Security Symposium. Denver: IEEE Press, 2000: 1-18.
[33] PETERSON D S, BISHOP M, PANDEY R. A flexible containment mechanism for executing untrusted code[C]// Proceedings of the 11th USENIX Security Symposium. San Francisco:CA Press, 2002: 207-225.
[34] PREVELAKIS V, SPINELLIS D. Sandboxing applications

[C]//Proceedings of USENIX Annual Technical Conference. Washington DC: IEEE Press, 2001: 119-126.

[35] SCOTT K, DAVIDSON J. Safe virtual execution using software dynamic translation[C]//Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC'02). Las Vegas: IEEE Press, 2002: 209-218.

[36] LIE D, THEKKATH C, MITCHELL M, et al. Architectural support for copy and tamper resistant software [C]// Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating System (ASPLOSIX). California: ACM Press, 2000:169-177.

[37] SUH G, CLARKE D, GASSEND B, et al. AEGIS: architecture for tamper-evident and tamper-resistant processing[C]//Processing of International Conference of Supercomputing (ICS'03). California: ACM Press, 2003: 160-171.

[38] CHEN B, MORRIS T. Certifying program execution with secure processors [C]//9th Hot Topics in Operating Systems. Lihue:IEEE Press, 2003: 1-15.

[39] INTEL CORPORATION. Intel trusted execution technology software development guide [R]. INTEL Publication, 2005.

[40] ADVANCED MICRO DEVICES. AMD64 virtualization: secure virtual machine architecture reference manual [R]. AMD Publication, 2005.

[41] SMITH J E, NAIR R. The architecture of virtual machines [J]. IEEE Computer, 2005, 38(5): 28-32.

[42] ROSENBLUM M, GARFINKEL T. Virtual machine monitors: current technology and future trends [J]. IEEE Computer, 2005, 38(5): 39-47.

[43] WHITAKER A, COX R S, SHAW M, et al. Rethinking the design of virtual machine monitors [J]. IEEE Computer, 2005, 38(5): 57-62.

[44] JIN H. Computer system virtualization-theory and application [M]. Beijing: Tsinghua University Press, 2008:1-26.

[45] PAYNE B D, CARBONE M, SHARIF M, et al. Lares: an architecture for secure active monitoring using virtualization [C]//Proc of the 29th IEEE symposium on Security and Privacy. Washington DC: IEEE Computer Society, 2008: 233-247.

[46] SHARIF M, LEE W, CUI W, et al. Secure in-VM monitoring using hardware virtualization[C]// Proc of the 16th ACM Conference on Computer and Communications Security. New York: ACM Press, 2009:477-487.

[47] GARFINKEL T, ROSENBLUM M. A virtual machine introspection based architecture for intrusion detection [C]//Proc of the 10th Network and Distributed System Security Symposium. Berkeley: USENIX Association, 2003: 191-206.

[48] SRIVATHS R, ANAND R. Security in embedded systems: design challenges [C]//Proceeding of the ACM Transactions on Embedded Computing Systems. New York: IEEE Press, 2004:461-491.

[49] KAPIL K. DRM on android[C]//Proceeding of the 10th IEEE India Conference. Mumbai:IEEE Press, 2013:1-6.

[50] TIAGO A, DON F. Trust zone: integrated hardware and software security enabling trusted computing in embedded system [J]. Government Information Quarterly, 2004, 3(4): 18-24.

[51] WAQAR A, DALE L. Redesigning secure element access control for NFC enabled Android smart phones using mobile trusted computing [C]//Proceeding of the 2013 International Conference on Information Society. Toronto: IEEE Press, 2013: 27-34.

[52] ARM LIMITED. ARM security technology building a secure system using trustzone technology [R]. ARM Publication, 2009.

[53] RUBY B L, PETER C S. Architecture for protecting critical secrets in microprocessors[C]//Proceeding of the 32nd International Symposium on Computer Architecture. Madison: IEEE Press, 2005:2-13.

[54] PETER W, ALEXANDRE. Implementing embedded security on dual-virtual-CPU systems[C]//Proceeding of the Design and Test of ICs for Secure Embedded Computing. Shilong:IEEE Press, 2007:582-591.

[55] YANG An, ZHAO Bo, LI Hongbo. Extension implementation of TCM in the embedded system based on FPGA[C]// Proceeding of the 2013 International Conference on Computer Science and Application. Wuhan: IEEE Press 2013:749-752

[56] DAVID J L, CHANDRAMOHAN T. Architecture support for copy and tamper-resistant software[C]//Proceeding of the ASPLOS-IX 2000. Massachusetts:IEEE Press, 2000: 1-10.

[57] EDWARD S, DWAINE C. AEGIS: architecture for tamper-evident and tamper-resistant processing [C]// Proceeding of the 27th International Conference on Supercomputing. San Francisco:IEEE Press, 2003:1-12.

[58] LI Hongjuan, LAN Yuqing. A design of trusted operating system based on Linux [C]//Proceeding of the 2010 International Conference on Electrical and Control Engineering. Wuhan:IEEE Press, 2010:4598-4601.

[59] ARIJIT U, JAYDIP S. Embedded security for Internet of things[C]//Proceeding of the 2nd National Conference on Emerging Trends and Applications in Computer Science. Shilong: IEEE Press, 2011:1-6.

[60] ARM LIMITED. ARM Trustzone API specification version 3.0 [R]. ARM Publication, 2009.

(编辑 王小唯)