XEN 虚拟机分析

薛海峰[1]，卿斯汉[2]，张焕国[1]
（1. 武汉大学计算机学院 2004 级博，湖北 武汉 430072；2. 中国科学院软件研究所，北京 100080）

摘 要： 硬件计算能力的极大提高重新激发了人们研究虚拟机软件的热情。XEN 虚拟机是目前业界广泛看好的一款开源的虚拟机管理软件，它具有良好的体系结构和优越的性能。随着虚拟机的应用，会出现新的安全问题。介绍了 XEN 虚拟机的系统结构，重点探讨了基于 Intel VT-x 的硬件虚拟技术，从处理器管理、内存管理和设备管理三个方面阐释了 **XEN 虚拟机的基本工作原理及其实现的关键技术**；在此基础之上，探讨了 **XEN 虚拟机的安全问题并提出了一些安全措施**。

关键词： 虚拟机；XEN；安全；VT-x
中图分类号： TP316.89 **文献标识码：** A **文章编号：** 1004-731X (2007) 23-5556-03

Analysis on XEN Virtualization Machine

XUE Hai-feng[1], *QING Si-han*[2], *ZHANG Huan-guo*[1]
(1. School of Computer, Wuhan University, Wuhan 430072, China; 2. Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: The great improvement of the hardware computing capability has stimulated people's enthusiasm to study virtual machine monitor again. XEN is an open-source virtualization machine monitor, whose future is widespread optimistic by the industry. It has a good system architecture and surprising performance. Companying With the application of the virtual machine, there are some new security challenges to be proposed. The hardware virtualization technology was interpreted based on Intel's VT-x technology and *the essences of XEN from the prospects of processor, memory and device managements were analyzed.* Based on the above analysis, the *security of XEN was discussed and some solution schema against security flaws was proposed.*

Key words: virtualization machine; XEN; security; VT-x

引 言

IBM 公司在上世纪六七十年代最早提出了虚拟机概念并将其运用到 VM/370 系统中。但是后来虚拟机发展一度停滞。现在硬件性能的大幅提升以及网络服务整合的需求，使得虚拟机技术获得了良好的发展基础和广泛的应用前景。目前的虚拟机管理软件有很多，例如：Bochs、Crusoe、QEMU、BIRD、VMWare、Virtual PC、UML 等，它们的一个共同问题是性能比较低，难以适应企业级应用。XEN 是剑桥大学教授 Ian 等发起的一个开源的虚拟机项目[1-2]，其性能接近单机操作系统(Native Operating System)的性能。由于其优越的性能和开源性，所以被业界广泛看好，被认为是未来最有前途的一款虚拟机管理软件。这里所讨论的虚拟机是系统虚拟机，即支持操作系统的虚拟机。

虚拟机具有很多优点和重要的应用场合，在许多文献[3-5]都有提到。下边列举其中一些：1、充分共享计算机资源，多个操作系统可以同时存在和运行于同一台计算机，操作系统的部署灵活方便，并且能够有效隔离操作系统和资源；2、当客户端操作系统（Guest OS）崩溃后恢复比较容易，开发特权软件将变得容易；3、能够虚拟物理上不存在的硬件；4、安全策略的实施管理和维护更加灵活和容易；5、容易实现操作系统的数据重放和回滚（Roll-Back）操作。

但是，虚拟机在具有诸多的优点和便利的同时，也出现一些新的安全性问题。本文第一部分主要介绍 XEN 虚拟机结构和实现原理以及硬件虚拟技术支持的 XEN 虚拟机；第二部分给出了性能测试；在分析了其关键技术基础上，最后一部分分析了 XEN 存在的安全问题，并提出了一些安全措施，总结和展望了未来 XEN 的发展。

1 XEN 虚拟机及 VT-x 的硬件虚拟技术

X86 体系结构设计之初没有考虑对虚拟机的支持，这为基于 X86 的虚拟技术带来一些问题和挑战，Robin 和 Irvine 分析了 X86 体系结构在虚拟化时的问题[6]。一些传统的系统虚拟机，例如：Bochs、Crusoe、QEMU、BIRD、VMWare、Virtual PC、UML 等，其实现复杂困难并且效率较低。目前 XEN 吸纳了各个虚拟机的长处和优势，使用了基于硬件的虚拟机技术，各虚拟机的大多数指令可以直接在处理器上运行，所以具有优越的性能和效率。

1.1 VT-x 的硬件虚拟技术

由于 X86 体系结构对虚拟技术的支持存在先天不足，Intel 公司发布了硬件虚拟机技术（Virualization Technology），

收稿日期： 2006-09-22 **修回日期：** 2006-12-05
基金项目：北京市自然科学基金(4052016)；国家自然科学基金(60673071, 60573042)；国家重点基础研究发展规划(973)(G1999035802)
作者简介： **薛海峰**(1975-)，男，河南人，博士生，研究方向为虚拟机技术，信息安全；**卿斯汉**(1939-)，男，湖南人，研究员，教授，博导，研究方向为信息系统安全理论和技术。

其中支持 X86 体系结构的 VT-x[7]。此后 AMD 也向外界发布了代号为 Pacifica 的虚拟化技术[8]（简称 SVM），以下本文主要针对 VT-x 技术。

Intel的X86CPU通过CPUID.1:ECX.VMX[bit 5]=1标识处理器支持VT-x技术。VT-x技术提供了两种CPU运行环境：根（Root）和非跟（Non-root），VMM（即XEN）运行于根环境，而各个虚拟机运行于非根环境。在根和非根之间可以转换，如果虚拟机要进行一些特权操作、I/O操作或者发生中断的时候，此时就进入了根环境，当处理完这些特权操作，重新进入非根环境继续虚拟机的执行。从根环境到非根环境叫VM Entry ；反之称为VM Exit 。VMM可以通过执行VMXON和VMXOFF指令打开和关闭VT-x。在Root和Non-root之间进行切换时，有一个虚拟机控制结构VMCS（Virtual Machine Control Structure）进行控制和管理，当虚拟机被创建时，VMM就同时为每一个VCPU（Virtual CPU）创建一个VMCS，这个数据结构可以决定哪些操作会触发VMExit进入Root。正是由于有了这个结构，VMM可以灵活地配置和管理虚拟机。

1.2 XEN 虚拟机结构及基本工作原理

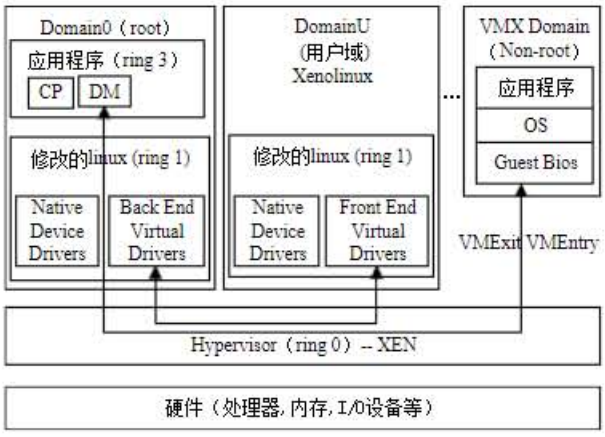

图 1 XEN 3.0 结构示意图，其中 CP 为控制面板（Control Panel），DM 为设备模型（Device Model）

XEN 在设计之初为了追求高性能，采用了半虚拟化（Para-Virtualization）技术，需要少量修改 Guest OS（客户端操作系统）内核与 VMM 协同工作，后来在加入了 Intel 和 AMD 的 VT-x 及 SVM 硬件虚拟技术支持后，XEN 能够实现全虚拟（Full-Virtualization）化，此时 Guest OS 内核不需要修改，Winndows XP 这种非开源的操作系统可以在其上运行，弥补了半虚拟化时的不足。从 XEN 3.0[2]开始就实现了具有硬件虚拟技术支持的全虚拟化。下边主要讨论 XEN 3.0。图 1 为 XEN 3.0 的结构示意图。其中 Domain（域）称为虚拟域，也就是虚拟机（Virtual Machine:VM）。XEN，即 Hypervisor，亦称为 VMM（Virtual Machine Monitor），直接运行于硬件裸机上，处于特权级 ring 0，用于控制管理和虚拟硬件资源，负责对各个虚拟机的调度以及对共享资源的控制访问等。Domain0 的内核运行在 ring 1，又称为 Xen0，它是一个具有特殊地位的虚拟机。控制面板（Control Panel: CP）作为一个特殊的应用程序通过 Hypercalls 来创建、保存、恢复、移植和销毁各个 Domain。设备模型[9]（Device Model: DM）为提供访问外设的途径。在 XEN 提供半虚拟的时候，用户域（DomainU：User Domain）上运行的 Guest OS 称为 XenoLinux（或者称为 XenU），特指内核修改过的 Linux。在全虚拟时，运行于其上的用户域称为 VMX Domain。

1.3 处理器管理

在 XEN 提供半虚拟的时候，VM 的内核是修改了的 Linux，运行于 ring 1。IA32 指令集包含了大约 250 多条指令，绝大多数指令可以直接在处理器上运行，但是其中的 17 条指令如果在 ring 1 运行将会遇到麻烦[6]，导致通用保护错（General Protection Fault：GPF）或者执行失败。对于这些问题指令（通常是一些特权指令），XEN 为虚拟机提供了 Hypercall 系统调用，通过 int 0x82 陷入（trap）指令实现，其类似于 Linux 的 int 0x80 系统调用，在 xen\arch\x86_32\entry.S 中定义了 32 位 x86 执行 int 0x82 陷入时的入口及相关工作。XEN 把这些特权指令用 Hypercall 替换，或者 XEN 监控某些特权指令，当发生 GPF 时，XEN 分析后进行相应处理，返回执行结果。通过 XenoLinux 和 XEN 的协同工作就可以正确执行特权指令。

在全虚拟时，由于有了 VT-x 支持，所以当 VM（运行于 Non-root）执行一些特权指令时，比如 I/O 访问、对控制寄存器的操作、MSR 的读写等指令，会导致处理器发生 VM Exit，此时处理器进入 root 环境，XEN 取得控制权，通过读取 VMCS 结构中的 VM_EXIT_REASON 字段得到发生 VM Exit 的原因，在 vmx_vmexit_handler 函数中开始执行相应处理。

目前 XEN 使用 BVT（Borrowed Virtual Time ）[10]调度算法。VM 获得一个时间片后连续运行于一个逻辑处理器，时间片耗尽后，XEN 调度下一个 VM 运行。

1.4 内存管理

内存的虚拟化是虚拟技术的一个难题。由于每个操作系统维护自己一套完整的内存访问和管理机制，包括 PDT、PTE、TLB 和 CR3 等数据，完成虚拟地址和物理地址间的翻译。在虚拟机系统中，各个虚拟机之间的内存访问要实现完全隔离，如果隔离不完全，就会发生灾难性的后果。在 XEN 系统中，XEN 统一管理机器实际的物理地址（现称为：机器地址，即机器实际拥有的内存地址空间），各个 VM 需要实现各自虚拟地址到物理内存地址（现称为：物理地址）间的转换。如图 2 所示。VMM 统一管理机器地址和物理地址的转换需要。XEN 有两种内存管理方式：直接方式（Direct Mode）和影子模式（Shadow Mode）。

在直接方式下，通过修改 Guest OS 内核中页表访问操作和 TLB 操作，Guest OS 就可以使用自己的页表直接访问

机器内存。PTE 或者 PDE 中的数据是物理内存地址，需要与机器地址相互转换。当 VM 执行页表访问时，执行 Hypercall 陷入（Trap）到 XEN，Hypervisor 帮助 VM 进行特权 MMU 操作，完成后将正确的地址返回。

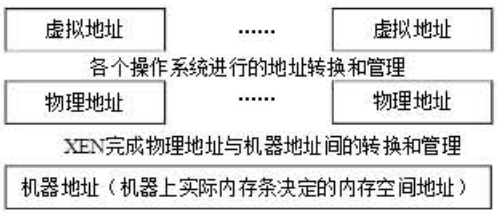

图 2 内存地址访问和管理示意图

在全虚拟时，Guest OS 内核是未经修改的，所以不能使用直接方式。此时为了完成物理地址与机器地址间的互相转换，VMM 使用影子模式对内存进行管理和虚拟。Guest OS 维护的页表中存放的是虚拟内存地址跟物理内存地址间的映射，为了完成访问实际的机器地址，VMM 要为每个 Guest OS 建立和维护影子页表，完成物理地址与机器地址间的映射，并及时做好这两张表之间的同步。各个 VM 进行页表操作时会导致 VM Exit，VMM 捕获这一事件，先让 Guest OS 更新其维护的页表，然后根据机器地址和物理地址的映射关系，用机器地址更新相应的影子页表项，从而完成内存访问。

1.5 设备管理

一个机器只有一套 I/O 地址和设备，设备管理和访问是操作系统头痛的问题，同样也是虚拟机实现的一个难题。另外虚拟提供虚拟设备供各个 VM 使用。设备管理同样要完成各个虚拟机间的设备隔离。

半虚拟时，XEN 为 VM 提供了一套设备抽象接口，各个 VM 传输 I/O 数据需要经过 XEN，通过共享内存页（Shared Memory Page）和步缓冲描述符环完成设备访问。在这个过程中，虚拟机通过前端虚拟驱动经过 XEN 与 Domain 0 的后端虚拟驱动进行交互，后端虚拟驱动再通过原始的设备驱动程序访问物理设备，从而完成设备的访问。

全虚拟时，XEN 通过移植 QEMU 的设备模型（DM：Device Model）进行设备管理和访问。Domain0 即 XEN0 提供了一个操作平台和设备管理模型的环境，它可以直接访问硬件设备，并为各个 VM 提供虚拟 I/O 服务。XEN 使用轻量级的事件通道（Event Channel）发送和接收异步通知，这些通知包括虚拟中断请求、物理中断请求以及域间的通信。为了提高设备访问的性能，通过共享内存页（Share page）来实现数据交换。这个共享内存页使用授权表（Grant-Table）来进行访问控制。下边以 NE2000 发读数据包说明设备访问。

(1) 当 VM 通过自己的网络驱动程序发送网络数据包时，IO 操作会触发 VM Exit，然后根据 VMCS 中的 VM_EXIT_REASON 判断发生 VM Exit 的原因，转入 VMExit 的处理函数进行处理。

(2) Hypervisor 将 I/O 指令的具体内容写入共享内存页（Shared Memory Page），它是由 VM 与 DM 共享，然后通过事件通道（Event channel）通知 Domain0。接着 Hypervisor 阻塞该虚拟机，并且调用调度算法，执行合适域。

(3) 当 Domain0 获得调度时，Hypervisor 恢复 Domain0 的状态，并把执行控制权交给 Xen0，此时开始执行 Xen0 的设备模型。回调函数 hypervisor_callback 调用 evtchn_do_upcall，evtchn_do_upcall 里收集有哪些虚拟机有多少 I/O 请求。此时 NE2000 的 I/O 请求同样被收集到 fd_set。

(4) DM 不停循环，通过 select 系统调用等待 I/O 请求，如果请求到达，DM 通过读取 I/O 共享页识别是对哪类外设访问，然后执行初始化时注册的相应的回调函数。对于 NE2000 来讲，就会通过 tun_receive_handler 函数调用虚拟 NE2000 网卡驱动程序的处理函数。

(5) 虚拟 NE2000 完成相当于硬件 NE2000 网卡的逻辑操作，通过 Xen0 修改过的驱动，就可以把 Guest OS 需要的网络数据放在了相应的 IO 地址。同时通过事件通道通知 Hypervisor 处理完毕。

(6) Hypervisor 得到通知后，解除对应的请求 I/O 的域的阻塞状态。当 VM 再次被调度，继续收发网络数据包。

I/O 中断的处理流程类似与上述 I/O 请求操作流程类似。原始的 QEMU 使用 Polling 方式轮询那个 I/O 唤醒，其效率较低，对于 NE2000 来讲，其速度不到 200KB/S，而采用事件方式后，可以达到 2MB/S。XEN 另外还提供了 PCNet 网卡的虚拟，其速度可以达到 7MB/S。在将来加入 VBD 技术后，磁盘性能会大幅提高。

2 性能测试

XEN 经过多种工具测试都有较理想的结果，比如 SPEC CPU2000、Lmbench、SPEC JBB2000、Cyclesoak、Sysbench 等，表 1 给出使用 SPEC CPU2000(表中简写为 CPU2k)进行测试的结果。使用的测试平台是：处理器是 Intel(R) Xeon(TM) CPU 3.00GHz，内存为 DDR2 533MHz 1G，使用单核，操作系统内核都为：Linux 2.6.16-rc5。

表 1 CPU2k 的在各个系统中测得的整型和浮点值性能数据

	CPU2k int	CPU2k fp
Linux-2.6.15-rc5	1951	1328
Linux-2.6.15-rc5-xen0	1903	1309
Linux-2.6.15-rc5-xenU	1938	1317
Linux-2.6.15-rc5-VMX	1885	1288

3 结论

目前的 XEN 的安全性还有较多的安全问题。Xen0 是一个安全瓶颈，其功能较其他域强，所以容易被敌手发起蠕虫、病毒、DoS 等各种攻击，如果 Xen0 瘫痪或者被敌手攻破，那么将破坏整个虚拟机系统。XEN 的隐通道问题没有解决，这样在 XEN 上就不可能运行高安全等级的操作系统。

（下转第 5569 页）

[2] Koichi S, Shoichi L, Hirohisa T. Development of Flexible Microactuator and Its Applications to Robotic Mechanisms [C]// Proceedings of the 1991 IEEE International Conference on Robotics and Automation. Washington, USA: IEEE, 1991, 4(2): 1622-1627.
[3] Gelenn K K, Joseph M C, Blake H. Artificial muscles: Actuators for Biorobotic Systems[J]. The international Journal of robotics research (S0278-3649), 2002, 21(4): 295-309.
[4] Deng W, Zhang H G . Fuzzy Neural Networks Adaptive Control of Micro Gas Turbine with Prediction Model [C]// Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control. Ft. Lauderdale, Florida, USA: IEEE, 2006: 1053-1058.
[5] Lin F J, Shieh H J, Huang P K. Adaptive Wavelet Neural Network Control With Hysteresis Estimation for Piezo-positioning Mechanism [J]. IEEE Transactions on Neural Networks (S1045-9227), 2006, 2(17): 432-444.
[6] 刘鸿文. 材料力学 [M]. 北京: 高等教育出版社, 2003.
[7] 吴家龙. 弹性力学 [M]. 上海: 同济大学出版社, 2001.
[8] El-Rabaie N M, Awad H A, Mahmoud T A. Wavelet fuzzy neural network-based predictive control system [C]// Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference, Dubrovnik, Croatia, IEEE, 2004, 1(12): 307-310.
[9] 孙炜, 王耀南. 模糊小波基神经网络的机器人轨迹跟踪控制 [J]. 控制理论与应用, 2003, 20(1): 49-53.
[10] 谭永红, 党选举, 梁峰. 基于动态小波神经网络的非线性动态系统辨识 [J]. 系统仿真学报, 2001, 13(s): 51-55. (Tan Y H, Dang X J, Liang F. Dynamic Wavelet Neural Network Based Nonlinear Dynamic System Identification [J]. Journal of System Simulation, 2001, 13(s): 51-55.)
[11] 朱娟萍, 侯忠生. 神经网络模糊非参数模型自适应控制及仿真 [J]. 系统仿真学报, 2006, 18(6): 1623-1625. (Zhu J P, Hou Z S. Fuzzy Non-Parameter Model Adaptive Control Method Based on Neural Networks and Simulations [J]. Journal of System Simulation, 2006, 18(6): 1623-1625.)
[12] 胡玉玲, 曹建国. 基于模糊神经网络的动态非线性系统辨识研究 [J]. 系统仿真学报, 2007, 19(3): 560-562. (Hu Y L, Cao J G. Research on Identification of Dynamic Nonlinear System Based on Fuzzy Neural Network [J]. Journal of System Simulation, 2007, 19(3): 560-562.)
[13] Hu Y L, Qiao J F. Fuzzy neural network control of uncertain parameters system[C]// The fifth World Congress on Intelligent Control and Automation, Hangzhou, China, IEEE, 2004: 2612-2616.
[14] Zhang H Y, Pu J X. A Novel Self-Adaptive Control Framework via Wavelet Neural Network [C]// The Sixth World Congress on Intelligent Control and Automation, Dalian, China, IEEE, 2006: 2254-2258.

（上接第 5558 页）

虚拟机的使用为数字版权保护提出新的挑战。虚拟机共享同一套硬件设备，一些网络安全协议可能更加容易遭到恶意破坏和恶意实施。此外，XEN 提供了方便的保存和恢复机制，使得操作系统数据的回滚和重放非常容易。这些将影响操作本身的密码特性。目前 XEN 本身的健壮性还有待完善，其资源隔离和共享访问控制等需要进一步的完善和加强。

消弱 XEN0 的功能，将其功能分解到其他域，将会适当减少 XEN0 的瓶颈作用。对于隐通道问题的分析和处理，是一个难解的问题，需要更多研究者的努力。为了资源隔离和资源共享安全，需要实行严格的访问控制策略。例如 IBM 将 sHyper 加入 XEN 中进行强制访问控制[11-12]；Intel 将要实施的 LT（LaGrande Technology ）技术，能有效增强设备隔离，实现 I/O 保护、内存越界保护、键盘、显示的隔离保护等。通过虚拟 TPM，建立可信域。另外对于虚拟机系统的管理需要特别的注意和加强。以上这些措施都可以有效增强 XEN 的安全性。

虽然目前 XEN 还有很多问题，但是由于其优越的性能、开源性和良好的架构，在全球各大公司，例如：Intel、AMD、HP、IBM 等的积极参与下，XEN 是业界最优秀的虚拟机之一。

致谢：感谢英特尔亚太研发有限公司的 OTC Linux VMM 小组，使我有幸参与 XEN 项目并且得到了大力的帮助，从而能够有机会完成此篇文章

参考文献:

[1] Paul Barham, Boris Dragovic, Keir Fraser, *et al*. XEN and the Art of Virtualization [C]// ACM Symposium on Operating Systems Principles (SOSP'03. ACM), ISBN: 1-58113-757-5. New York, USA: ACM Press, 2003: 164-177.
[2] Ian Pratt, Keir Fraser, Steven Hand, *et al*. XEN 3.0 and the Art of Virtualization [EB/OL]. (2005) [2007]. http:// www. linuxsymposium. org /2005/linuxsymposium_procv2.pdf.
[3] Susanta Nanda, Tzi-cker Chiueh. A Survey on Virtualization Technologies [EB/OL]. (2005) [2007]. http:// www. ecsl. cs. sunysb.edu/tr/TR179.pdf.
[4] P.M. Chen, B.D. Noble. When Virtual Is Better Than Real [C]// Proceedings of Workshop on Hot Topics in Operating Systems, 2001(III). Elmau, Germany. USA: IEEE Computer Society Press, 2001: 133-138.
[5] Nadir Kiyanclar. A Survey of Virtualization Techniques Focusing on Secure On-Demand Cluster Computing [EB/OL] (2005) [2007]. http://citeseer.ist.psu.edu/kiyanclar05survey.html.
[6] J. S. Robin, C. E. Irvine. Analysis of the Intel Pentiums Ability to Support a Secure Virtual Machine Monitor [EB/OL] (2000) [2006]. http://citeseer.ist.psu.edu/robin00analysis.html.
[7] Intel Corporation. Intel® Virtualization Specification for the IA-32 Intel Architecture [EB/OL]. (2005) [2006]. http://cache-www.intel.com/cd/00/00/19/76/197666_197666.pdf.
[8] AMD Corporation. AMD64 Virtualization Codenamed 'Pacifica' Technology Secure Virtual Machine Architecture Reference Manual [EB/OL]. (2005) [2006]. http: // _www.amd.com /us-en /assets /content_type/white_papers_and_tech_docs/33047.pdf.
[9] Keir Fraser, Steven Hand, Rolf Neugebauer, *et al*. Safe Hardware Access with the Xen Virtual Machine Monitor [EB/OL]. (2004) [2006]. http: //www. cl.cam.ac.uk/ Research/ SRG/ netos/ papers/ 2004-oasis-ngio.pdf.
[10] Kenneth J. Duda, David R. Cheriton. Borrowed-Virtual-Time (BVT) scheduling: supporting latency-sensitive threads in a general-purpose scheduler [C]// Proceedings of the 17th ACM SIGOPS. Symposium on Operating Systems Principles, volume 33(5) of ACM Operating Systems Review. New York, USA: ACM Press, 1999: 261-276.
[11] Reiner Sailer, Trent Jaeger, Enriquillo Valdez, *et al*. Building a MAC-Based Security Architecture for the XEN Open-Source Hypervisor [C/OL]. 21st Annual Computer Security Applications Conference. December, 2005. Tucson, Arizona. (2005) [2006]. http://www.acsac.org/2005/papers/171.pdf.
[12] Sailer R, Valdez E, Jaeger T, *et al*. sHyper ： Secure Hypervisor Approach to Trusted Virtualized Systems [EB/OL]. (2005) [2006]. http:// domino.watson.ibm.com/ library/ cyberdig.nsf/ 3addb4b88e7a231f85256b3600727773/265c8e3a6f95ca8d85256fa1005cbf0f?OpenDocument&Highlight=0,Hypervisor.