Artigo: Architectural Support for Secure Virtualization under a Vulnerable Hypervisor [1]
Autor do Resumo: Thiago Augusto Lopes Genez (RA 100616)

Resumo

A computação em nuvem emergiu como um novo paradigma, onde hardware e software são entregues como serviços de utilidade geral e disponibilizados para os usuários através da Internet. Graças à camada de virtualização, construída sobre os recursos físicos, é possível otimizar estes recursos com o uso de máquinas virtuais – *Virtual Machine* (VM). Em outras palavras, a virtualização é a tecnologia que abstrai as características físicas do hardware e fornece recursos virtualizados para as aplicações de alto nível. Assim, uma máquina física é, de modo geral, logicamente dividida entre um monitor de máquina virtual – *Virtual Machine Monitor* (VMM) – e várias máquinas virtuais. Entretanto, cada máquina virtual é independente (e isoladas) uma das outras, podendo ter seu próprio sistema operacional, aplicativos e serviços. O VMM, também conhecido como *hypervisor*, é responsável pelo controle e virtualização dos recursos físicos compartilhados entre as VMs, como por exemplo, processadores, memória RAM, disco rígido e dispositivos de entrada e saída. É na vulnerabilidade do compartilhamento da memória RAM que os autores Jin *et al.* descreveram em [1].

Segundo Jin *et al.*, o *hypervisor* geralmente possui um controle total sobre os recursos de hardware, podendo acessar, sem nenhuma restrição, a memória RAM virtualizada de qualquer VM. Ao comprometer o *hypervisor*, um usuário malicioso pode acessar o conteúdo da memória das VMs usadas pelos usuários da nuvem. Isto é, a propriedade de *isolamento* das VMs é quebrada. Então, Jin *et al.* propõem um mecanismo baseado em hardware para proteger a memória RAM das VMs de acessos não autorizados. Denominado *hardware-assisted secure virtual machine* (H-SVM), este mecanismo independe da confiabilidade do *hypervisor*. Em outras palavras, os autores retiraram a responsabilidade de a alocação e gerenciamento da memória RAM física do *hypervisor* (software), e a transferiram para o hardware. No entanto, o mecanismo proposto pode ser vulnerável a ataques de hardware (não remotos), tais como leitura da memória RAM após desligá-la. Os autores, porém, assume que o provedor de nuvem é confiável e não tem a intenção de comprometer os hardwares.

Um dos principais requisitos importantes para o H-SVM é minimizar as modificações necessárias na arquitetura do hardware atual e no *hypervisor*, pois estas mudanças seriam muito complexas. Ou seja, o mecanismo H-SVM deve ser integrado nos projetos de processadores atuais sem aumentar significamente a complexidade do hardware, ou seja, o custo extra de hardware deve ser pequeno. Além disso, as interfaces com o *hypervisor* também deve ser simples. Resumidamente, o mecanismo H-SVM funciona da seguinte maneira: para quaisquer alterações na alocação de memória para uma VM, o *hypervisor* solicita um pedido ao H-SVM para atualizar a memória física. Assim, com intuito de impedir visualizações de memória RAM de qualquer VM pelo *hyperviosr* ou usuário malicioso, o H-SVM verifica se a solicitação pode violar o isolamento de memória entre VMs. Jin *et al.* realizaram experimentos do mecanismo H-SVM no *hypervisor Xen*, e os resultados mostraram que o *Xen* tornou-se seguro contra ataques na memória RAM das VMs e, além disso, o desempenho do *hypervisor* não foi prejudicado, ou seja, o *overhead* (custo extra) embutido no hardware foi mínimo e satisfatório.

Portanto, o mecanismo H-SVM melhora o isolamento de memória entre VMs através do bloqueio das alterações diretas na memória física pelo *hypervisor*, sendo gerenciado apenas pelo hardware. Assim, o provedor de nuvem pode oferecer um ambiente de execução seguro às VMs dos seus clientes.

Referências

[1] S. Jin, J. Ahn, S. Cha, and J. Huh, "Architectural support for secure virtualization under a vulnerable hypervisor," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44 '11. New York, NY, USA: ACM, 2011, pp. 272–283. [Online]. Available: http://doi.acm.org/10.1145/2155620.2155652