Paper: CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization
NCS 490 - Virtualization
Instructor: Ronny Bull
Student: Thomas McCombie
3/5/2013
This paper, ‘CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization’ describes a privileged virtual machine that is used to help secure commonly available hypervisors. This paper describes the variety of attack vectors, both external and internal, and how CloudVisor is used to mitigate these risks. In addition, the authors do some benchmarking of a system using CloudVisor versus a vanilla instance of Xen.

The idea that hypervisors are insecure comes from the privileges that exist between guest VM, virtual machine monitor, and management software. These components can be considered attack surfaces. A compromise in one of these attack surfaces can be used as a jumping point to attack another. Ultimately, an attacker may be able to pull or manipulate data from a completely different virtual machine, or disrupt an entire hypervisor. Many cloud providers are using commonly available hypervisors to reduce cost and increase customer compatibility. In doing so, they unwittingly expose their customers to these attack vectors. Inherently secure hypervisors can and have been developed, however, they have proven to be unpopular and not used. To encourage secure use, CloudVisor was developed as a kind of bolt on to increase the security of commonly available hypervisors.

CloudVisor acts much like intrusion prevention sensor. It operates at the most privileged level, and demotes the virtual machine monitor to a guest level. In doing so, it acts as a conduit for information traveling between the virtual machine and the virtual machine monitor. By doing so, it can control what memory should be controlled by which virtual machine, provide encryption capabilities to disk I/O, and access denial if a virtual machine tries to make an illegal request.

Because virtual machines have to share the same physical memory, there is the potential that one guest virtual machine could access the memory space of a different virtual machine. This means an advisory could gain access to a victim’s information by first gaining access to a neighboring virtual machine and using it as an attack vector. This is troubling because no matter how good you secure your OS, your actual security is only as good as your weakest peer virtual machine guest. CloudVisor does a variety of checks, including remapping memory locations, clearing memory locations upon reset, and using bounce buffers and “trap and emulate” requests to ensure integrity.

Disk I/O is protected by CloudVisor through the use of encryption. CloudVisor utilizes both symmetric and asymmetric cryptography to accomplish this, thus protecting data from both external and internal attacks. Additionally, data requests are verified before being passed to the guest virtual machine, thus ensuring integrity. However, these additional steps are not without risks, because these steps delay disk writes, a sudden power failure can potentially cause corruption to a virtual disk. However, the authors of this paper assume a robust battery backup environment. Because asymmetric encryption is used, it is necessary to facilitate key transfers. This is especially important during virtual guest migrations between hypervisors.

Because of the additional translation layer, performance is reduced. The amount of reduction is dependent upon the activity of the virtual machine. Disk I/O is significantly reduced due to the encryption operations being performed. Certain memory operations can be significantly slowed down due to the number of reset interrupts needed for CloudVisor to operate.
I found the concept of this security method interesting. It allows the user to continue to use familiar hypervisors, yet bolt on additional security features to further isolate each virtual machine. However, the performance degradation can be significant and I believe would hinder it’s acceptance into the market. However, what I would hope for is that some of the checks and security features of this product would be integrated into the virtual machine monitor itself. This would lead to some performance increases. Additionally, they could be made optional. However, it’s not without notice that the authors stated that separating out these security operations from the VMM would allow developers to continue to concentrate solely on each of their focused projects. As a side note, I am aware that the DOD has sponsored several efforts to help harden virtualization platforms. In all, I found the concept interesting and worth further refinement.