Seguridad en Entornos Virtuales

Javier Echaiz* Jorge Ardenghi

Laboratorio de Investigación de Sistemas Distribuidos (LISiDi)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Bahía Blanca (8000), Argentina
{je,jra}@cs.uns.edu.ar

Resumen

En un entorno virtual, un *monitor de máquina virtual* (VMM) controla múltiples VMs mediante una abstracción de software del hardware subyacente. Esta arquitectura provee algunas ventajas con respecto a la seguridad pero también introduce desafíos únicos. Irónicamente los avances en la potencia de cómputo y la disminución de los costos del hardware fueron los factores que dieron origen a la pérdida de interés en la virtualización, hoy principales contribuyentes de su renacimiento.

La virtualización surgió a finales de la década de 1960, con el objetivo de multiplexar las aplicaciones sobre mainframes de forma tal de poder repartir los escasos y costosos recursos de cómputo entre múltiples procesos. La creación de las VMs hizo posible que múltiples aplicaciones coexistiesen sobre una máquina única.

Esta línea de investigación busca desarrollar nuevas tecnologías centradas en la seguridad de los entornos virtuales, especialmente a nivel del VMM.

Palabras Clave: Virtualización, Detección de Intrusos, Seguridad en Redes, Políticas de Seguridad, Sistemas Distribuidos, Computación Colaborativa, Automatización.

1. Introducción

Cuarenta años después del surgimiento de la virtualización, modernos sistemas operativos (SOs), altas velocidades en los procesadores y bajos costos de hardware resolvieron los problemas para los que se inventaron las VMs. Fue posible entonces instalar nuevas aplicaciones tan fácilmente y barato como instalarlas en un servidor dedicado con su propio procesador, memoria y almacenamiento. Sin embargo, esta tendencia dio origen a nuevos problemas: el hardware barato hizo que proliferasen las máquinas subutilizadas, las cuales a su vez ocuparon espacios significativos y aumentaron el overhead en su administración. Mantener las aplicaciones actualizadas y los patches de los sistemas operativos en cada server se volvió una tarea pesada. Lograr que éstas máquinas fuesen seguras implicó que se determinasen claramente las responsabilidades de las organizaciones sobre cada server. Por ende, mover aplicaciones nuevamente a las VMs dentro de unas pocas máquinas físicas y administrándolas mediante monitores de VMs (VMMs) se volvió una solución factible.

Un VMM es una capa de software que usualmente corre directamente sobre el hardware. En los sistemas actuales el VMM puede correr junto a un sistema operativo host. Las VMMs exportan una abastracción de máquina virtual que se asemeja al hardware subyacente. Cada abstracción de máquina virtual es un *guest* que encapsula el estado completo del SO que corre dentro de ella. El sistema operativo guest interactúa con la abstracción virtual del hardware, la cual es a su vez manejada por el VMM como si fuese hardware real. La VMM es entonces un "sistema operativo de sistemas operativos". La VMM usualmente corre en modo privilegiado mientras que el sistema operativo guest lo hace en modo usuario.

*Becario CONICET, Argentina.

2. Máquinas virtuales

Básicamente existen dos tipos de entornos de máquinas virtuales. En el primero, el VMM se conoce con el nombre de *hypervisor* y se implementa directamente entre el hardware y los sistemas guest. El ejemplo clásico de este tipo de máquinas virtuales es el hypervisor de Xen (http://www.xen.org). Este tipo de virtualización es especialmente aplicable a entornos de servidores tipo UNIX.

En el otro tipo de monitor de máquina virtual, el VMM corre como una aplicación más dentro del SO guest y depende de él para prover drivers de I/O y código *bootstrap* en lugar de implementarlo directamente desde cero. Este tipo de VMM se popularizó en PCs comunes (siendo posiblemente VMware (http://www.vmware.com) el más difundido), cuya plataforma x86 no fue originalmente diseñada para soportar completamente la virtualización por hardware. Este tipo de entorno suele ser considerado menos seguro que el anterior debido a que el VMM depende de los servicios provistos por el sistema operativo host.

La virtualización actual se ha desviado de sus raíces originales como una herramienta de multiplexado para convertirse actualmente en una solución para hacer frente a los problemas de seguridad, confiabilidad y administración. Este hecho presenta consecuencias positivas y negativas, pues por un lado la virtualización junto con sus técnicas asociadas ayudan a resolver múltiples problemas de seguridad, especialmente porque las máquinas virtuales corren sobre un único sistema, el cual puede implementar un sistema seguro multinivel con sistemas virtuales separados en cada nivel. Por otro lado la virtualización fabrica espacios para nuevas vulnerabilidades, donde los mecanismos de seguridad tradicionales no están preparados para resolver estos problemas. Esta línea de investigación se suma entonces a la búsqueda de nuevos mecanismos para combatir estos nuevos vectores de ataque.

2.1. Monitores de máquinas virtuales (VMMs)

Los VMMs soportan tres atributos en entornos virtuales [1]: aislamiento, interposición e inspección. Los VMMs propician el aislamiento debido a que las VMs no comparten memoria física. Gracias a la abstracción de la memoria virtual el VMM puede crear la ilusión de que cada VM tiene su propio espacio de direcciones. De esta forma cada VM corre sin saber de la existencia de otras VMs, pues todas sus acciones están confinadas a su propio espacio de direcciones.

Para soportar interposición los VMMs gestionan todas las operaciones privilegiadas a nivel del hardware físico. Los SOs guest transfieren todos los traps e interrupciones al VMM para procesar los eventos. El VMM intercepta todos los pedidos de I/O provenientes de los dispositivos virtuales de las VMs y los mapea al dispositivo físico de I/O correspondiente. Gracias a esta abstracción el VMM gestiona y planifica todas las VMs simultáneamente. Los sistemas operativos guest no saben de la existencia del propio VMM, ni saben que se encuentran compartiendo recursos con otras VMs; por el contrario la abstracción les presenta la ilusión de estar interactuando directamente con los dispositivos físicos.

Por último los VMMs tienen acceso a todos los estados de una VM, incluyendo el estado del CPU, de la memoria y de los dispositivos. Esta visión provee las capacidades de inspección, propiciando *checkpoints*, *rollbacks* y *replays*. Adicionalmente estos atributos se combinan para brindar otra característica deseable: *portabilidad*. Los administradores pueden guardar, copiar, mover e instanciar entornos con estados encapsulados desde una máquina a otra.

Por todo ello decimos que el VMM es en sí mismo un pequeño sistema operativo que atrapa los eventos que surgen en los sistemas operativos guest, gestionando sus I/Os y mapeando datos a memoria y discos virtuales. Al mismo tiempo la implementación física está separada y oculta de los SOs guest pero colectivamente encapsulada permitiendo replicación y migración eficiente.

2.2. Virtualización: lo bueno

Las características inherentes de la virtualización y los atributos de las VMMs que arriba resumimos simplifican la gestión de los recursos de cómputo. La reducción del overhead administrativo facilita el proceso de hacer seguro un sistema.

2.2.1. Pool de hardware

Los VMMs brindan una vista uniforme del hardware subyacente y por ello una plataforma de hardware puede contener múltiples entornos virtuales. Por lo tanto los administradores pueden ver a una computadora como parte de un pool de recursos de hardware genéricos [2]. Esta característica permite disminuir costos de hardware y bajar los requerimientos de espacio. Adicionalmente, dado que el VMM puede mapear máquinas a recursos de hardware disponibles se simplifica el balance de carga y la escalabilidad, volviendo triviales las fallas de hardware.

2.2.2. Encapsulamiento

La propiedad de encapsulamiento mejora la seguridad desde varios frentes. Las VMs son fácilmente encapsulables y replicables, por lo tanto los administradores pueden sistemáticamente agregar en tiempo de ejecución nuevos servicios y aplicaciones al entorno replicado.

Cada nuevo servicio o aplicación puede correr independientemente sin el riesgo de corromper o interferir con otros. Si un sistema cae (*crash*) u ocurre un ataque, los administradores pueden suspender a la VM afectada, hacer un rollback a un estado de ejecución estable y recomenzar. Por último, analizar mediante replay puede exponer configuraciones defectuosas o proveer información acerca del método de ataque, vulnerabilidades, etc.

2.2.3. Logs seguros

Los servicios de seguridad, como los logs seguros y la protección de intrusos a nivel del SO puede implementarse también en el VMM. El hecho de que estos servicios se ejecuten separados de todos los procesos en un entorno virtual mejora sus capacidades.

La implementación de logs seguros a nivel del SO tiene la desventaja de que un atacante puede deshabilitar o modificar los logs una vez comprometido el sistema y por ello tampoco los logs proveen información útil para un efectivo análisis de seguridad forense. Estos problemas se solucionan con la implementación a nivel del VMM, pues el atacante no puede modificar los logs y con ello se vuelven una herramienta útil para el análisis forense.

Además la virtualización mejora la prevención y detección de intrusos mediante el uso de *clones*. Los sistemas de detección de intrusos suelen basarse en firmas de patrones de ataques conocidos, tornándolas prácticamente inútiles frente a nuevos eventos sospechosos. Por otro lado los IDSs basados en anomalías corren el riesgo de marcar como ataques acciones legítimas; o peor aún pueden aceptar eventos maliciosos repetidos como actividad normal. Los clones por otra parte pueden correr eventos sospechosos capturados en una copia virtual del sistema real y observar los cambios sin comprometer el sistema real.

Por último, la protección a intrusos en el nivel virtual permite detecciones imposibles para los sistemas tradicionales, pues permiten monitorear todos los eventos que ocurren en el entorno virtual y por lo tanto hacer cumplir una política preestablecida detectando por ejemplo *cracking* de passwords a partir de una lectura de bloques de discos que contienen un archivo con passwords seguido de una gran actividad de CPU.

2.2.4. VMs en cuarentena

La virtualización permite que los administradores puedan poner una o más VMs en cuarentena (fuera de la red) y buscar vulnerabilidades, evidencias de ataques, prevención de diseminación de código malicioso hacia otros nodos, etc.

2.2.5. Distribución de software

Para la mayoría de los sistemas complejos, las posibles combinaciones de hardware, versiones de sistemas operativos y librerías vuelven impráctico que los desarrolladores de software contemplen cada posible combinación. La virtualización alivia estos problemas permitiendo que los desarrolladores distribuyan máquinas virtuales completas que contengan su software.

Los dispositivos portables de almacenamiento flash, por ejemplo *pen drives*, permiten extender aún más este concepto. Los usuarios suelen emplear estos dispositivos para llevar consigo documentos, imágenes, etc. Con la virtualización es posible que un usuario lleve consigo una copia de su VM junto a sus archivos de trabajo y llevar consigo su máquina completa en un bolsillo.

2.3. Virtualización: lo malo

En la subsección anterior vimos ventajas de seguridad a partir del uso de entornos VM: infraestructuras que automáticamente realizan balance de carga, detectan fallas independientes de hardware que hacen que las VMs migren, se creen y se destruyan según la demanda de servicios particulares. Sin embargo, algunas propiedades de la virtualización hacen que lograr un nivel de seguridad aceptable en entornos VM sea más difícil.

2.3.1. Proliferación de VMs

La escalabilidad constituyó uno de los puntos fuertes desde la creación de los entornos virtuales, pues crear una nueva VM es tan simple como encapsular y generar mediante copia una nueva instancia. Sin embargo ésto no siempre es bueno, pues los usuarios pueden tener demasiadas VMs de propósitos especiales, e.g., una para testing, otra para demostraciones, otra como *sandbox* para probar nuevas aplicaciones, y otra por cada sistema operativo. Esta situación torna inmanejable el tema desde el punto de vista de la performance y desde el gasto considerable de memoria física.

Además, desde el punto de vista de la administración la situación se vuelve compleja, pues las actualizaciones, configuraciones, etc. deben hacerse en cada máquina. Desde la perspectiva de la seguridad, la proliferación de VMs puede abrumar a los administradores, haciendo inseguras a algunas VMs (exponiendo a amenazas a la organización).

Finalmente la virtualización imposibilita algunos procedimientos tradicionales de seguridad, por ejemplo dado que múltiples VMs corren en el mismo host físico, deshabilitar un *port* en la máquina host para hacer segura una aplicación especial en una VM específica presenta el indeseable efecto de deshabilitarlo también para el resto de las VMs que podrían necesitarlo.

2.3.2. Comportamiento transitorio

El beneficio de la portabilidad es a la vez un problema grave de seguridad. Dado que las VMs se replican fácilmente puede aparecer una legión de "máquinas transitorias" que aparecen y desaparecen de la red. Esto puede llevar a que vulnerabilidades que no existían ahora aparezcan brevemente, infecten otras máquinas y luego desaparezcan antes de detectarlas, complicando en gran medida el trabajo de los administradores de la seguridad, no sólo respecto a la gestión de patches sino también dificultando los *penetration tests*.

Si bien los checkpoints, rollbacks y replays de las VMS ayudan a recuperar fallas en entornos virtuales también presentan el problema de la reexposición a vulnerabilidades, reactivación de servicios riesgosos, rehabilitación de cuentas desactivadas, etc.

2.3.3. Aspectos sociales

Los aspectos sociales relacionados con la virtualización son sociales por naturaleza. La facilidad de instanciación de las VMs puede llevar a que los administradores simplemente remuevan y reinstalen una VM comprometida en lugar de que analicen lo sucedido en un ataque.

2.3.4. Nuevos riesgos

Los entornos virtuales traen aparejados nuevos riesgos, como una máquina virtual robada que alguien llevaba en un pen drive. Éste es un riesgo análogo al de una laptop robada pero en este caso con un dispositivo de menor tamaño y por ende más fácil de sustraer. Adicionalmente un cracker podría tratar de robar máquinas virtuales completas atacando *file servers*.

3. Línea de investigación

En el presente artículo se analizaron brevemente algunos aspectos a tener en cuenta a la hora de emplear virtualización como alternativa segura a las máquinas físicas. Vimos que no todas son ventajas, y por ello nos proponemos investigar en profundidad en este campo, área que resurgió a partir de las promesas de una seguridad mejorada, confiabilidad y ventajas administrativas.

En particular nuestro interés se centra en la implementación de servicios de seguridad avanzados a nivel del VMM, explotando vistas que le son imposibles a un sistema de seguridad a nivel del SO guest.

Las promesas de la virtualización arriba mencionadas son viables pero debemos ser concientes de los riesgos inherentes que plantea esta tecnología, pues si bien es cierto que se disminuyen overheads, se facilita la administración y se combaten las vulnerabilidades de seguridad a nivel del SO, no es menos real que se introducen nuevos riesgos que van en detrimento de la seguridad del sistema.

Referencias

[1] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *NDSS*, The Internet Society, 2003.

[2] M. Rosenblum and T. Garfinkel, "Virtual machine monitors: Current technology and future trends," *IEEE Computer*, vol. 38, no. 5, pp. 39–47, 2005.

[3] P. M. Chen and B. D. Noble, "When virtual is better than real," in *HotOS*, pp. 133–138, IEEE Computer Society, 2001.

[4] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: Security challenges in virtual machine based computing environments," in *Proceedings of HotOS X: The 10th Workshop on Hot Topics in Operating Systems*, USENIX, June 2005.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R.Ñeugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, (Bolton Landing, NY, USA), ACM, Oct. 2003.

[6] R. Sailer, T. Jaeger, E. Valdez, R. Perez, S. Berger, J. L. Griffin, and L. van Doorn, "Building a MAC-based security architecture for the Xen open-source hypervisor," in *21st Annual Computer Security Applications Conference*, ACM, Dec. 2005.

[7] S. Weber, P. A. Karger, and A. Paradkar, "A software flaw taxonomy: aiming tools at security," *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–7, 2005.