



## ¿La virtualización de aplicaciones virtualiza también su seguridad?

El término "Virtualización" es uno de esos términos vagos que tanto gustan y que están tan solicitados dentro de las campañas de marketing de productos, por otra parte, netamente "concretos"<sup>1</sup>. Este vocablo se refiere a la posibilidad de abstraerse de los recursos concretos del ordenador en el que estamos ejecutando las aplicaciones. Por "recursos" debemos entender cualquier elemento del ordenador, todo (discos, pantalla, teclado, tarjeta de red, sistema de ficheros, etc.).

El proceso de abstracción que conlleva la virtualización persigue reducir el contenido de información de cada componente operativo y quedarse sólo con lo que es relevante para su propósito particular. Dicho de otro modo, si lo que necesitamos es acceder a un disco duro, el sistema de virtualización nos "muestra" un disco duro pero sin reparar en detalles como de qué tipo es, qué hardware está detrás, si se trata de un solo disco o son varios, si ese disco es realmente un disco duro o una memoria CompactFlash<sup>2</sup>, si existe o si está realmente dentro de la máquina (virtual) que nosotros, en tanto que software que se ejecuta, vemos. En este contexto, la virtualización es cualquier técnica que oculte las características físicas concretas de los recursos de un sistema de cómputo, frente a las aplicaciones y usuarios finales que la utilizan.

El término virtualización fue utilizado profusamente

**Últimamente se habla mucho de las arquitecturas dirigidas a servicios, de la virtualización de entornos, de la distribución de cómputos y almacenamientos, pero realmente no se está tratando cómo afectan estos modelos a la seguridad de los sistemas. Puede que la virtualización no sea tan buena como pueda parecer, a la hora de saber qué se está ejecutando en el sistema y quién realmente los está operando. Quizás suponga cambios de geometría y nuevas necesidades de comunicación que afectan a la seguridad de todo el sistema.**

en la década de los sesenta o incluso antes, y se ha hecho en diferentes aspectos de la computación que van desde todo el ordenador hasta alguna de sus funcionalidades o componentes. Lo único que tienen en común todos los usos del término es la ocultación de los detalles técnicos a través de la *encapsulación*<sup>3</sup>.

La virtualización crea un interfaz externo que oculta la implementación concreta que hay detrás; un ejemplo de ello puede ser las granjas de servidores web que desde

a los cambios en el diseño o la implementación; esta abstracción proporcionaría interfaces estables.

### La virtualización de plataforma y la virtualización de recursos

Al igual que ocurre con los términos *abstracción* y *orientado a objetos*, la virtualización es una palabra que se utiliza en diferentes contextos, y que tiene significados distintos en cada caso. Los dos enfoques

lización en los años sesenta se refería concretamente a las *máquinas virtuales* (o *pseudo-máquinas*) creadas mediante la combinación de software y hardware adecuados. Por su parte, el término *máquina virtual* se remonta a los tiempos del sistema experimental IBM M44/44X<sup>4</sup>.

La virtualización de la plataforma se realiza sobre un hardware concreto y mediante el uso de un software corriendo en el *anfitrión* (programa de control), que crea un entorno computacional simulado (una máquina virtual) para el software invitado.

Este software invitado suele ser un sistema operativo completo y se ejecuta como si estuviese instalado él sólo, aislado, en la mencionada plataforma hardware. Hay numerosas y diferentes formas de hacer esto y se distinguen en el grado de



**La gran inestabilidad de la virtualización es que, una vez corrompida o restaurada la única copia de referencia, los efectos perniciosos o correctivos de esta acción se distribuyen diligentemente a todos los usuarios.**

fuera parecen ser solo uno y en realidad puede haber decenas o centenares de ellos detrás. Otro ejemplo lo encontramos en los volúmenes lógicos construidos, de modo transparente, sobre diferentes discos físicos. En principio, la ocultación de los detalles de diseño o de implementación está justificada en tanto que proporcione inmunidad frente

más significativos son la **Virtualización de Plataforma** y la **Virtualización de Recursos**; en la primera se intenta abstraer de los detalles del sistema de cálculo subyacente, y en la segunda lo que se quiere ocultar son los detalles de cómo se implementan uno o varios recursos del sistema (almacenamiento, memoria, etc.) El origen del término virtua-

la simulación del hardware: Emulación o Simulación, Virtualización Nativa o Completa, Virtualización por Hardware, Virtualización Parcial, Paravirtualización, Virtualización a nivel de Sistema Operativo, etc., pero a nosotros la que nos interesa ahora es la que se conoce como **Virtualización de Aplicaciones**.

La Virtualización de Aplicaciones consiste en poder ejecutar una aplicación de escritorio o un servidor de forma local, utilizando recursos locales, y poder hacerlo dentro de una máquina virtual apropiada; esto es distinto a ejecutar la aplicación como un software

<sup>1</sup> Algo es virtual cuando es visible, cuando es perceptible por el observador, pero no existe físicamente en la forma que se percibe, mientras que algo es real cuando es una parte existencialmente autónoma del mundo real.

<sup>2</sup> La CompactFlash fue inicialmente desarrollada como sistema de almacenamiento para artefactos digitales portables y típicamente utiliza memoria flash (no-volátil) como elemento esencial. La primera especificación y lanzamiento de este producto lo hizo SanDisk Corp. en 1994.

<sup>3</sup> En ingeniería del software la encapsulación se consigue mediante la separación de funciones y ocultación de la información con enfoques que permiten meter los elementos de programación dentro de entidades más abstractas y generalistas.

<sup>4</sup> Se basaba en una máquina modelo IBM 7044 y simulaba múltiples copias de sí misma en la forma de máquinas 7044 virtuales.

local convencional, es decir, como software que ha sido instalado en el sistema local. Tales aplicaciones virtualizadas corren en un pequeño entorno en el que sólo aparecen los componentes necesarios para su ejecución (entradas en el Registro de aplicaciones, ficheros, variables de entorno, elementos de la interfaz con el usuario, objetos globales y locales, etc.). Este entorno virtual actúa como una capa entre la aplicación y el sistema operativo, y elimina los conflictos que pueda haber entre aplicaciones y los que éstas puedan tener con el sistema operativo. Ejemplos de este proceder los encontramos en productos como la **Java Virtual Machine** de Sun, en el *SoftGrid*<sup>5</sup> de Microsoft, en los productos de Altiris (de Symantec), por mencionar algunos. Algunos ejemplos de uso los encontramos en la *consolidación de servidores (physical-2-virtual o transformaciones P2V)*, en los *sistemas para recuperación de desastres* (entornos *hot standby* que sustituyen a la filosofía *backup & restore*), los *sistemas de prueba y entrenamiento*, las *aplicaciones portables* (encapsulado de aplicaciones con redireccionamiento a ficheros temporales), los *espacios de trabajo portables* (como iPods y memorias USB), etc.

### Application Streaming

Un procedimiento relativamente novedoso<sup>6</sup> en este escenario es lo que se conoce como **Application Streaming**, que es una forma muy útil de distribuir software dentro de la virtualización de aplicaciones. En este paradigma, la aplicación a ejecutar proviene de un servidor que se encarga de enviar al cliente bloques de código ejecutable, y lo hace de modo que la aplicación puede comenzar a ejecutarse incluso antes de que se haya completado la descarga.

Este modo de operación es opuesto al modelo basado en terminales donde la aplicación se ejecuta en el servidor y se visualiza en el terminal. Una



**En los sistemas virtualizados la velocidad de propagación del malware es cero o infinita, es cuántica, sin medias tintas.**

primera diferencia importante entre ambos es que el tamaño del servidor necesario para dar servicio a un determinado número de clientes es mucho menor si se utiliza *Application Streaming*. Esta técnica hace que la aplicación se ejecute en los clientes y hace innecesarias las granjas de servidores y, además, las aplicaciones se ejecutan mucho más deprisa con lo que la impresión del usuario final es mucho mejor.

Hay dos modelos distintos de *Application Streaming* en el mercado: **1)** el de empaquetar juntas cierto número de aplicaciones y enviarlas (*sandboxing*), y **2)** el de poner una capa transparente de software entre el sistema operativo, el registro de aplicaciones, y las aplicaciones, de modo que éstas puedan ser enviadas desde el exterior y luego ser ejecutadas como si estuviesen instaladas localmente.

### Analizar la seguridad con cuidado

La seguridad de estas soluciones hay que analizarla con cuidado. En realidad, la virtualización sólo altera un poco la ubicación de los mismos elementos de siempre. En todos los sistemas, virtualizados o no, las aplicaciones a ejecutar están almacenadas en alguna parte, por lo que éstas pueden corromperse del mismo modo, independientemente de que estén almacenadas en los discos

locales de cada cliente, o lo estén en los discos del servidor de aplicaciones.

El hecho de que una versión virtualizada no tenga acceso de

escritura sobre el fichero de referencia de esa aplicación, no significa que eso no se hiciese ya con una adecuada política de seguridad en los tradicionales sistemas multi-usuario. En el caso virtualizado, esa copia está en el servidor de aplicaciones y sería allí donde debería dirigir sus pasos el atacante.

### Inestabilidad de la virtualización

La gran inestabilidad de la virtualización es que, una vez corrompida o restaurada la única copia de referencia, los efectos perniciosos o correctivos de esta acción se distribuyen diligentemente a TODOS los usuarios. En los sistemas virtualizados la velocidad de propagación del *malware* es cero o infinita, es cuántica, sin medias tintas.

Además de esto, ya es tiempo de ir pensando en otros tipos de ataques en los que el atacante no necesita corromper y almacenar una versión de su software dentro del sistema. Imaginemos un escenario en el que el código maligno se agazapa en un servidor web, en una URL inocente y muy visitada, y logra transferirse al navegador "virtualizado" de un usuario; si este navegador tiene algún fallo explotable, el código malicioso quizás pudiese entrar en ejecución en esa misma instancia del navegador y empezar a hacer de las suyas mientras nosotros seguimos confiados haciendo lo que estuviésemos haciendo. Al

apagar el navegador, es verdad que nada quedaría almacenado para futuras invocaciones, pero quizás el mal ya esté hecho. Si el código atacante consigue actuar como usuario del navegador, aquel podrá hacer las mismas operaciones que haga el usuario a la vez que él las hace (enviar a otros sitios la contraseña que acaba de entregar a su legítimo web bancario) o mientras el usuario hace otras cosas si esos valores sensibles están en el entorno del navegador (ficheros históricos, de contraseñas, de direcciones, de identidades digitales, etc.)

La virtualización realmente no cambia nada, ni mejora nada, en lo que se refiere a la *integridad de lo que se está ejecutando* (software contaminado), o a *quien lo está realmente operando* (suplantación e inyección de comandos no deseados por el usuario). La virtualización de las aplicaciones puede ser muy efectiva a la hora del control de la calidad del software de partida (análisis antivirus), a la hora de las actualizaciones y correcciones, o a la hora de necesitar una flexibilidad de gestión máxima cuando el número de clientes es muy grande pero, realmente, no cambia nada esencial en el *Juego de Guerra* que supone el ataque y la defensa de los sistemas software.

En principio, la virtualización de las aplicaciones y de los escritorios es una capa más de software, pero **sí afecta a la estabilidad de todo el sistema** ya que con ella se vuelve a **geometrías centralistas** en las que la protección de los datos, de todos los datos, puede ser más difícil, y en la que los servicios centrales se convierten en una perita en dulce para el atacante, a la par que **umentan mucho los "beneficios" de cualquier ataque.** ■

JORGE DÁVILA MURO  
Consultor independiente  
Director  
Laboratorio de Criptografía  
LSIIS – Facultad  
de Informática – UPM  
jdavila@fi.upm.es

<sup>5</sup> *SoftGrid* es una aplicación de virtualización y de *application streaming* de Microsoft, como consecuencia de la adquisición de la empresa bostoniana *Softricity* en julio de 2006. Ésta no es la única herramienta de virtualización que tiene Microsoft, por lo que quizás sea una adquisición para que "no le hagan sombra a sus productos nativos".

<sup>6</sup> Ver los productos y patentes de las compañías *Omnishift* y *StreamTheory* desde el año 2000.