SISTEMAS OPERATIVOS (2017-2018)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Maquinas Virtuales

Antonio Rodríguez Alaminos

19 de noviembre de 2017

Índice general

Capítulo 1

Introducción

1.1. ¿Qué es Virtualización?

Se podría definir Virtualización como la tecnología que permite ejecutar varios sistemas operativos o máquinas virtuales en una misma máquina física. Esta tecnología puede ser implementada por hardware o por software, siendo esta última más versátil, flexible y extendida. [5]

En una computadora donde se virtualice van a coexistir dos tipos de sistemas operativos:

- Sistema operativo anfitrión: Es el que manejará el hardware, el que deberá tener los controladores adecuados para interactuar con los componentes instalados. En este se definirá qué recursos serán asignados a cada una de las máquinas virtuales.
- Sistema operativo huésped: Son los sistemas operativos que correrán dentro de las máquinas virtuales definidas en el servidor. Cada uno de estos sistemas operativos interactúa con el hardware físico a través de dispositivos virtuales generado en una capa de virtualización gestionada por el sistema operativo anfitrión. Los recursos de hardware asignados a cada máquina virtual podrán ser modificados según las necesidades, asignándole más memoria, mayor almacenamiento, o bien, quitándole o agregándole interfaces de red. Estas modificaciones podrán realizarse con la máquina virtual apagada o encendida dependiendo de las capacidades de la solución de virtualización implementada.

Teniendo en cuenta que cada sistema operativo asume que tiene el control del hardware sobre el que corre, es evidente que para lograr que los sistemas operativos puedan convivir se requiere de un servidor de virtualización que administre el acceso a la memoria, el uso del CPU y por supuesto las entradas/salidas. Este servidor es una

capa de software que interactúa con el sistema operativo anfitrión y con los sistemas operativos huéspedes.

1.2. Tipos de Virtualización

Los sistemas de virtualización se pueden clasificar entendiendo tres paradigmas distintos: los sistemas de virtualización completa (Full Virtualization), los sistemas de paravirtualización y los que se conocen como virtualización a nivel de sistemas operativo:

- Full Virtualization: Es el paradigma más utilizado hoy en día. Los sistemas operativos huéspedes desconocen estar siendo virtualizados, por lo tanto no requieren ser modificados. Se utiliza un hipervisor entre las máquinas virtuales y el hardware. Los sistemas operativos huéspedes realizan las peticiones directamente al hardware virtualizado y ante cualquier instrucción que requiera acceso privilegiado el hipervisor la intercepta para su apropiado manejo. Esta arquitectura de virtualización facilita la migración de máquinas virtuales entre diferentes sistemas, e incluso entre diferentes arquitecturas. Esta flexibilidad y versatilidad tiene aparejado una penalización en la performance y el uso de recursos. Ejemplos:

  - VMWare
  - Qemu-KVM
  - VitualBox

- Paravirtualización: Esta tecnología implica una modificación en el sistema operativo huésped que incorpore cambios en los llamados a drivers evitando la necesidad del hipervisor de hacer de proxy entre el sistema operativo huésped y el hardware. En estos sistemas cada máquina virtual actúa con el hardware de manera similar a cualquier otro proceso del sistema. Ejemplos:

  - Xen
  - LDoms

- Virtualización a nivel de Sistema Operativo: Este sistema es muy diferente a los dos anteriores. Este tipo de virtualización permite a un sistema operativo crear múltiples entornos de aplicaciones aislados que referencian al mismo kernel. Una ventaja importante que tienen es que su penalización es muy baja. Ejemplos: AIX partitions, Solaris containers. [6]

1.3. Procesos y Modos de ejecución

Un proceso es un programa en ejecución. Tradicionalmente, un proceso en Linux tiene dos modos de ejecución: kernel y usuario. El modo usuario (user mode) es un

modo de ejecución sin privilegios comúnmente usado por aplicaciones de usuario, o sea cualquier código que no pertenezca al kernel.

El modo kernel (kernel mode) es solicitado cuando una aplicación requiere servicios del kernel, como puede ser escribir a disco. Ante la necesidad de ejecutar una operación de E/S o crear un nuevo proceso hijo, el proceso en modo usuario debe realizar una llamada al sistema para solicitar la acción que requiere ejecutar.

1.4. Tipos de control de accesos

El control de accesos es un mecanismos de protección interno que tienen los sistemas operativos multiusuario para proteger los recursos de accesos indebidos.

Unix/Linux históricamente ha usado el esquema de control de acceso discrecional (DAC). Donde generalmente el propietario del recurso o archivo en cuestión, administra los derechos de acceso sobre el mismo y también puede especificar a cual de sus grupos pertenece el recurso. [4]

Sumado a esto, debemos saber que un proceso en ejecución toma los derechos de acceso que tiene el usuario que lo ha invocado. Así un proceso puede modificar o borrar un archivo en particular, si el usuario que lo invoca cuenta con los derechos de acceso suficientes en el recurso. [2]

Siguiendo esta lógica es que el kernel decide si un proceso puede o no leer, escribir o ejecutar un recurso en particular. Aunque debemos saber también que esta lógica es usada para usuarios normales, el super usuario root no sigue esta lógica, ya que es el dueño del sistema operativo y el tiene acceso completo a todos los recursos del sistema.

En contraste con lo anterior existe el control de acceso mandatorio (MAC), implementado en sistemas open sources a través del kernel SELinux. En este tipo de control de acceso cada objeto (archivos o procesos) tiene una etiqueta que representa el nivel de sensibilidad de la información que contiene. Adicionalmente cada usuario tiene asociado un nivel de permisos que indica a que tipos de objetos puede acceder. Estas autorizaciones son especificadas en las políticas SELinux determinado ademas que permisos tendrá al hacerlo. [2]

Capítulo 2

Seguridad.

2.1. Seguridad en Virtualización Open Source

La seguridad en virtualización debe ser estudiada desde un punto de vista un poco más amplio que al pensarlo para un servidor sin virtualización. Una brecha de seguridad en una máquina virtual podría traer problemas en otras máquinas virtuales del mismo host físico o incluso podría comprometer el mismo host anfitrión. Así mismo si el host anfitrión es inseguro todas las máquinas virtuales serán vulnerables.

Cuando se analiza la seguridad de un hipervisor se deben tener en cuenta los riesgos introducidos por la plataforma de virtualización, que generalmente estarán relacionados con:

- El nivel de aislamiento entre máquinas virtuales, tanto a nivel de red como de entrada/salida;
- La protección de los accesos a las consolas de las máquinas virtuales, antes física, ahora remota;
- El aseguramiento de los accesos a la gestión y administración del hipervisor.

El presente análisis y sus recomendaciones han sido pensados para escenarios cada vez más comunes en la actualidad, en los que un servidor de virtualización alberga servidores de diferentes zonas de seguridad, por ejemplo: servidores de DMZ, servidores de Intranet y servidores que atienden a Extranets de Partners todos corriendo dentro de un mismo servidor físico.

Como se puede prever la convivencia de diferentes zonas de seguridad aumenta la complejidad de administración y gestión del servidor de virtualización. Por este motivo, todas las configuraciones para aumentar la seguridad deben ser planificadas, analizadas, implementadas y por supuesto documentadas. La documentación

de todos los esquemas y configuraciones resultan elementales para una administración cómoda y eficiente.

Además vale destacar que todas las recomendaciones y configuraciones son pensadas para usarse en Linux Debian Squeeze, aunque estas puede ser adaptadas para usarse en la mayoría de las distribuciones Linux.

Siendo tan amplio y disperso los temas que conciernen al análisis, para abarcar la problemática de seguridad en virtualización en su conjunto, esta se la analizará desde diferentes ángulos:

- La red: se analizará la interacción de las máquinas virtuales y el host anfitrión con el mundo. Se debe tener en cuenta que la proximidad y apareamiento de las máquinas virtuales entre sí, y con el host anfitrión es muy grande. Por lo tanto, el host anfitrión debería ser configurado de modo de aislar las máquinas virtuales de su propia red.

- El almacenamiento: se analizará el manejo de los datos y programas almacenados en medios físicos que usará cada máquina virtual y la propia del host anfitrión. Se debe asegurar el aislamiento de estos datos definiendo exactamente qué puede acceder cada proceso de kvm y restringiendo lo demás, para evitar posibles accesos indebidos.

- El acceso de administración: la creación, modificación y baja de hardware virtual, así como el encendido, migración y apagado de estas máquinas virtuales es una tarea de administración que debe ser asegurada mediante procesos de autentificación y autorización que impidan posibles ataques al hipervisor.

- El acceso a la consola de las máquinas virtuales: Debido a que las terminales de las máquinas huéspedes son consolas virtuales, se debe estudiar la manera mas conveniente de como restringir y controlar el acceso remoto a estas interfaces.

2.2. Asegurando la red

A continuación se detallarán todos los temas relacionados con la disponibilidad, integridad y confidencialidad de los datos que viajan por la red en un host anfitrión y por supuestos en sus huéspedes.

2.3. Alta disponibilidad de red

Al incrementar la densidad de servicios que dependen de una misma interfaz de red, esta comienza a ser más crítica y se recomienda tomar ciertos recaudos. Esto sucede en la interfaz de red que brinda conectividad a las máquinas virtuales que corren dentro de un mismo host, el aumento de la criticidad de esta interfaz amenaza la

disponibilidad de red para todas máquinas virtuales que se albergan. Por este motivo, es recomendable disponer de dos interfaces de red dedicadas para brindar la conexión a todas las máquinas dentro de un mismo equipo.
Para mayor facilidad de administración y una respuesta inmediata de alta disponibilidad lo conveniente es "unir" las interfaces bajo una virtual de modo que esta brinde servicio siempre que, al menos una de las interfaces físicas pueda hacerlo. Esta funcionalidad en linux se llama bonding y para el presente esquema el modo que usaremos es active-backup de modo que sólo una de las interfaces físicas atenderá en un momento y la otra será su backup.
Adicionalmente, es recomendable disponer de una tercera interfaz de red conectada a la red de Management o Administración que será utilizada exclusivamente para gestionar la máquinas virtuales de manera remota.

2.4. Más de una zona de seguridad virtualizada

En lo relativo a aislar diferentes zonas de seguridad partiremos del supuesto que la separación de estas, a nivel de red, lo tenemos implementado mediante VLANs. Partimos de esta base debido a que esta configuración es la más común en la actualidad.
Lo primero que debemos hacer es que el servidor de virtualización vea el conjunto de las VLANs de los zonas de seguridad tagueadas en protocolo IEEE 802.1q en ambas interfaces de servicio (eth0 y eth1). Para esto se deben configurar el switch al que están conectadas las interfaces para que en estas bocas se presenten las VLANs correspondientes.

Evitando ataques basados en técnicas de Spoofing ARP En caso de que una máquina virtual sea comprometida, esta podría ser utilizada para hacer un escalamiento horizontal del ataque. Muchos ataques se basan en el cambio de identidad de host atacante alterando la MAC address de la interfaz de red. Esta técnica es conocida como Spoofing ARP.
Para evitar esto lo que se recomienda hacer es definir reglas de firewall a nivel de capa física (capa 2 del modelo OSI). El objetivo de estas reglas será filtrar todo los paquetes que provengan de una interfaz virtual cuya MAC address no corresponda con la asignada al definir la máquina virtual.

2.5. Las máquinas virtuales como procesos

Dan Walsh, líder de desarrollo de SELinux en Red Hat, afirma que correr múltiples sistemas operativos en un mismo hardware "es una de las cosas que más miedo dan" desde el punto de vista de la seguridad informática[3]. Previo a la virtualización un host podía ser atacado a través de la red, pero si ahora varios sistemas corren sobre un mismo hardware, la "superficie de contacto" podría ser mucho mayor.
Como se detalla al comienzo del trabajo, Linux ve y trata a cada máquina virtual

como un proceso más, este tiene asociado los privilegios del usuario que lo invocó. Ahora, supongamos que se descubre un nuevo bug que posibilita la ejecución de código en el hipervisor desde una máquina virtual. Solamente pensar en las consecuencias que esto podría tener da escalofríos. Al respecto se pueden tomar algunas medidas que se desarrollan a continuación:

- Un usuario no privilegiado por cada máquina virtual
- sVirt el módulo de SELinux

2.6. sVirt el módulo de SELinux

El servicio sVirt es un modulo incluido por SELinux para aislar la máquinas virtuales entre sí y brindar un esquema de control de acceso mandatorio (MAC) [1]. Con esta funcionalidad se definen etiquetas únicas, de manera dinámica, para cada proceso que sea una máquina virtual reduciendo los accesos autorizados para cada máquina sólo a sus recursos asignados. De este modo, si una máquina virtual es comprometida, aún pudiendo escalar por sobre la virtualización para ejecutar código en el host anfitrión, las posibilidades de explotar esta situación serán nula.

Lamentablemente esta funcionalidad aún no está madura, ajustada y pulida en Debian Linux, ya que las pruebas realizadas no han sido exitosas.

Los inconvenientes encontrados al activar SELinux fueron:

- Presenta un error de falta de permisos al intentar asociar la interfaz túnel al bridge, esto sucede aun iniciando la máquina virtual con el usuario root.
- Luego de iniciada la máquina virtual (sin red por el problema anterior) los discos y el proceso en cuestión, no son asignados con las etiquetas que deberían.

Contrastando los resultados obtenidos con los manuales de Red Hat sobre el tema, es evidente que el desarrollo liderado por Dan Walsh no ha sido compartido en su totalidad con la comunidad o esta no la ha incorporado aún a los repositorios públicos. Encriptando la unidad de almacenamiento de los huéspedes. La encripción del disco es una técnica que se suele usar para proteger los datos almacenados en una unidad cuando esta no está en uso o cuando el dispositivo está apagado. Es muy común la encripción de discos en dispositivos portátiles para proteger la confidencialidad de los datos ante un posible extravío.

En el escenario de un hipervisor y sus máquinas virtuales esta técnica podemos usarla para proteger los datos de las máquinas virtuales mientras estas estén apagadas. Supongamos que un atacante logra acceder al hipervisor, si la máquina virtual está apagada los datos de la misma se encontrarán encriptados y por lo tanto inaccesibles. Estos tipos de ataques son conocidos como “Ataques offline” [1].

2.7. Administración remota del hipervisor

El acceso remoto para administración del hipervisor debe ser analizado en detalle, ya que es un punto crítico dentro de cualquier esquema de virtualización. Su criticidad radica en que, en caso de ser descuidado, un acceso no autorizado amenazaría toda la infraestructura virtualizada.

Dentro de los posibles esquemas de seguridad para el acceso de administración y gestión de máquinas virtuales se detallarán tres formas de acceso remoto [1]:

- Usando túneles SSH
- Usando autentificación SASL y encripción
- Usando TLS con llaves asimétricas

2.8. Administración remota a través de túneles SSH

Este modo de acceso remoto aprovecha la seguridad implementada en el protocolo de Secure Shell (SSH). Utiliza la conexión SSH como un túnel, que asegura integridad y confidencialidad dentro de la red, para comunicarse con el hipervisor de manera remota.

2.9. Administración remota usando autentificación SASL y encripción

El protocolo SASL es un estándar usado por múltiples aplicaciones que provee autentificación y encripción en conexiones. El servicio define un archivo de base de datos con las credenciales autorizadas. Además puede ser usado en escenarios más complejos usando servicios de autentificación externa como Kerberos o LDAP [1]

2.10. Administración remota usando TLS con llaves asimétricas

Las conexiones TLS (Transport Layer Security) basan su seguridad en encripción basada en certificados x509. Estos certificados son claves asimétricas firmadas por una autoridad certificantes (CA).

Para generar los certificados existen diversas formas, la más básica usando directamente el comando openssl. Pero para evitar que los certificados queden en claro en el disco usaremos las aplicaciones modutil y certutil que brindan un entorno para la generación y almacenado de certificados de manera segura.

2.11. Administración remota de la consola de las máquinas virtuales

Un punto importante en la virtualización es que desaparecen las consolas puras de las máquinas virtualizadas y estas pasan a ser virtuales también. Qemu-KVM brinda estas interfaces sobre el protocolo VNC para acceso remoto. Por defecto esta opción está deshabilitada y deberá estudiarse en detalle previo a publicar el video de las máquinas virtuales.

El acceso a la consola remota deberá ser protegido de dos maneras: exigiendo una autentificación mediante contraseña y habilitando el acceso al este puerto únicamente desde la estaciones de trabajo de los administradores.

2.12. Protegiendo el acceso con contraseña

Como primera medida para asegurar el acceso a la consola de las máquinas virtuales disponemos del uso de una contraseña en el servidor VNC que brinda acceso a dichas terminales. Aunque esto es seguro no es la solución óptima, ya que la limitación del largo de las contraseñas es 8 (ocho) caracteres haciendo que esta contraseña sea potencialmente vulnerable a ataques de diccionario. [1] Por este motivo debemos elegir una contraseña fuerte que incluya mayúsculas, minúsculas, números y caracteres especiales.

2.13. Protegiendo el acceso usando TLS y claves asimétricas

Para habilitar el acceso remoto vía VNC se recomienda hacerlo de modo que este brinde autentificación TLS con certificados x509. Para configurar esto podemos reutilizar los certificados que hicimos para el acceso a la administración del hipervisor. En caso de que quienes administren las máquinas virtuales y el hipervisor sean personas diferentes será importante que los certificados usados sean diferentes. Para generar nuevos certificados se puede seguir los pasos 1 al 6 de creación de certificados explicado en el apartado “Administración remota usando TLS con llaves asimétricas”.

Capítulo 3

Conclusión

Como se detalló a lo largo del trabajo, las variantes de configuración posibles son muchísimas. Esto exige a los administradores de este tipo de hipervisores un compromiso mayor con la seguridad ya que, como se vio, un error en la configuración podría comprometer la confidencialidad, integridad y disponibilidad de los datos y servicios alojados en el hipervisor y en las máquinas virtuales.

También es imprescindible conocer los nuevos riesgos introducidos con KVM para poder minimizarlos o llevarlos a un nivel aceptable para nuestra infraestructura.

Aunque las herramientas a nivel de red y administración remota están preparadas para dar un soporte acorde a las necesidades de seguridad actuales, quedo evidenciado en las pruebas realizadas que todavía a nivel de procesos y privilegios de usuario del hipervisor hay muchos interrogantes por responder.

Según la teoría y los manuales de Red Hat Enterprise, SELinux brinda una solución importante a través del modulo sVirt, que daría las respuestas pendientes en Debian Linux. De cualquier modo, un desarrollo más a conciencia y sin grandes cambios del módulo de administración de las máquinas virtuales puede permitirnos correr cada máquina virtual con un usuario diferente y de ese modo obtener un aislamiento más que aceptable sin caer en un modelo de control de acceso mandatorio.

Sin dudas es en este punto dónde aún queda mucho camino por recorrer, el presente trabajo no es más que un puntapié para un análisis más en profundidad sobre distintas alternativas de control de acceso y aislamiento de procesos que ayuden a minimizar los nuevos riesgos.

El sistema de virtualización estudiado es muy “joven” y está en pleno desarrollo, hoy en día impulsado por ser el sistema de virtualización recientemente elegido por Red Hat, desplazando su anterior solución Xen. Debido a esto la comunidad que desarrolla Open Source posiblemente será altamente beneficiada del código que pueda brindar Red Hat.

Más allá de los esfuerzos de los desarrolladores de la comunidad y de Red Hat, la responsabilidad de velar por la seguridad de los servidores que administremos es nuestra. Por este motivo, es muy importante mantener los hipervisores actualizados al día y seguir atentamente los reportes de seguridad. En este caso, DSA (Debian Security Announce) es el más indicado y en el que encontraremos todas las novedades al respecto.

Por experiencia propia puedo asegurar que además de las configuraciones iniciales recomendadas a lo largo del trabajo, para maximizar la seguridad es fundamental la correcta documentación y revisión de los procedimientos para adaptarlos a este nuevo paradigma.

Bibliografía

[1] IBM Blueprint: KVM Security First Edition. 2011.

[2] IBM Blueprint: KVM Tunning KVM for performance. 2011.

[3] Linuxcon: Secure virtualization with sVirt. 2011.

[4] Kernelthread.com: A Taste of Computer Security by Amit Singh. 2012.

[5] Fernando Picouto Jacinto Grijalba Maikel Mayan Ángel García Eduardo Inza y Carlos Alberto Barbero Antonio Ángel Ramos, Jean Paul García-Morán. Instala, administra, securiza y virtualiza ENTORNOS LINUX, 2009.

[6] Trent R. Hein y Ben Whaley Evi Nemeth, Garth Snyder. Unix And Linux System Administration Handbook, 2011.