Buenas prácticas de seguridad en entornos IaaS

Índice de contenido

1. Introducción

Este documento es de carácter técnico, y está destinado a personas con los suficientes conocimientos informáticos como para administrar sistemas IaaS (Infraestructura como Servicio) conectados a Internet basados en GNU/Linux.

El presente documento es una guía básica de buenas prácticas para mantener la seguridad de sistemas IaaS en la Universidad de Sevilla. Este documento no es en ningún caso exhaustivo, y por tanto debe entenderse como *el* **conjunto mínimo de medidas a adoptar** para mantener razonablemente seguros los sistemas.

La seguridad de sistemas de Información se basa en tres pilares fundamentales:

1. *Medidas técnicas.*
2. *Medidas procedimentales.*
3. *Concienciación y formación.*

Si se descuida alguno de los pilares la seguridad se derrumba. Suele decirse que una cadena es tan fuerte como su eslabón mas débil, y esto es perfectamente aplicable a la seguridad de sistemas información.

Por poner un ejemplo, de nada sirve que un sistema genere alertas ante ataques si nadie revisa dichas alertas, o si dicha persona no entiende el significado de dichas alertas.

Este documento se focaliza en medidas de seguridad de sistemas servidores IaaS, pero no por ello han de olvidarse otras medidas de seguridad básicas concernientes a los puestos de trabajo / dispositivos móviles desde los que se accede a dichos sistemas IaaS.

2. Sistemas IaaS RHEL/Rocky Linux

Los sistemas IaaS basados en Red Hat Enterprise Linux o Rocky Linux son muy comunes en entornos universitarios. Se trata de sistemas de bajo o nulo coste de adquisición, de alto rendimiento y con una amplia selección de software disponible.

A lo largo de los siguientes puntos se darán una serie de buenas prácticas para mantener estos sistemas de la forma más segura posible.

2.1 Actualizaciones de seguridad.

Mantener actualizados los sistemas es una de las principales tareas que podemos realizar para incrementar su seguridad. A medida que se descubren problemas de seguridad en el software, éste se debe actualizar para corregir dicha vulnerabilidad.

El primer paso que se debe realizar es suscribirse a la lista de distribución de avisos de seguridad del sistema operativo usado, sea Red Hat o Rocky Linux, para recibir por correo electrónico los avisos de nuevas actualizaciones en cuanto estas sean liberadas.

Lista oficial de avisos de seguridad de Red Hat:

http://www.redhat.com/mailman/listinfo/rhsa-announce

Lista oficial de avisos de seguridad de Rocky Linux:

https://lists.resf.org/mailman3/lists/rocky-announce.lists.resf.org/

Para aplicar actualizaciones se usará la conocida herramienta **yum.** Se puede consultar la guía de actualizaciones (tanto para RHEL como para Rocky Linux) en el manual oficial de Red Hat Enterprise Linux disponible aquí:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-yum.html#sec-Updating_Packages

Es importante aclarar una serie de conceptos relativos a las actualizaciones:

- En el caso de servicios de red (ej: *apache, mysql, ssh, tomcat*) es necesario reiniciar el servicio asociado tras la actualización para que ésta se aplique.
- Si se actualizan librerías es necesario relanzar todos los servicios de red que estén enlazados con dichas librerías. Así una actualización de *openssl* o *glibc* debe llevar aparejada un reinicio de servicios como *apache*, *ssh* o *mysql*.
- Si el paquete a actualizar es el del *kernel*, es necesario reiniciar el sistema completo para que la actualización tenga efecto al arrancarse con el kernel actualizado.

Idealmente, las actualizaciones de seguridad se deberían aplicar en cuanto hubiese una actualización disponible, de ahí la importancia de estar suscrito a las listas de distribución de notificaciones del fabricante. En cualquier caso, se recomienda que se apliquen actualizaciones de seguridad al menos una vez por semana.

2.2 Eliminación de software y servicios innecesarios.

Un principio básico de la seguridad de la información es presentar una *"superficie de ataque"* lo más pequeña posible. Mientras mas paquetes de software tengamos instalados en nuestro sistema IaaS, mayor es la probabilidad de que aparezcan problemas de seguridad en dichos paquetes. Por tanto, deberían eliminarse aquellos paquetes que no necesitemos para prestar servicio en nuestro sistema.

El mismo principio de *"menor superficie de ataque"* es también aplicable a los servicios de red. Se deberían ejecutar únicamente los servicios estrictamente necesarios para la funcionalidad deseada del sistema.

Por ejemplo, un servidor web no tendría porque tener ejecutándose servicios como *nfs*, *portmap* o *rpc.* Del mismo modo, un servidor que ejecute cálculos científicos con *Matlab* o *Mathematica* no debería ejecutar servicios como *apache* o *mysql*.

Con el comando **nestat -apn | grep -i LISTEN | grep -i tcp** se pueden enumerar la listas de servicios TCP ejecutándose para así decidir cuales de ellos deben ejecutarse y cuales no. Habria que ejecutar el comando análogo con servicios basados en UDP.

Para detener y deshabilitar servicios no deseados estos son los comandos a ejecutar como usuario *root*:

Sistemas RHEL 7 / CentOS 7 /RHEL 8 / Centos 8 / Rocky Linux 8 :

systemctl stop nombre_servicio
systemctl disable nombre_servicio

Sistemas RHEL 5 / CentOS 5 / RHEL 6 / Centos 6:

service nombre_servicio stop
chkconfig –level 35 nombre_servicio off

2.3 Visibilidad de los servicios de red a prestar.

A la hora de prestar servicios de red es conveniente analizar las necesidades de visibilidad de dichos servicios. ¿Necesitamos que sean visibles desde toda Internet, y por tanto desde todo el planeta, o podríamos restringir el acceso a un subconjunto de direcciones?

A veces surge la necesidad de lanzar servicios para grupos de trabajo como podría ser una intranet con *Drupal*. El acceso a dicha intranet podría restringirse vía apache a los rangos de red de la Universidad de Sevilla y con usuario/contraseña como paso previo desde rangos externos.

Un caso muy paradigmático es el del motor de bases de datos *mysql*. Algunos administradores requieren acceso desde Internet a dicho servicio, siendo esto poco recomendable. Es mucho más razonable lanzar *mysql* en *localhost* y emplear un túnel *ssh* para conectarse desde remoto al servicio de base de datos[1].

En el caso de que sea necesario tener servicios de cara a toda Internet, recomendamos el uso del software *fail2ban* para reducir el impacto de ataques de fuerza bruta en servicios como *openssh* o múltiples autenticaciones erróneas desde una misma dirección IP en el registro de log de *apache*.

1 *El conocido software de gestión mysql workbench soporta nativamente túneles ssh para conectar a un servidor MySQL.*

2.4 Empleo de protocolos cifrados.

Se debe evitar el uso de protocolos no cifrados en aquellos servicios que transmitan datos confidenciales o autenticaciones (como usuarios y contraseñas). Por ejemplo, se debe evitar el uso de *ftp* (y usar en su lugar alternativamente *scp/sftp*) y toda autenticación web debe realizarse necesariamente empleando el protocolo seguro *https*.

A la hora de configurar el cifrado en las aplicaciones se evitarán protocolos obsoletos como *SSLv2* o *SSLv3* y se deben emplear los algoritmos de cifrado más fuertes posibles. Por ejemplo, para el conocido servicio Apache el siguiente cuadro es una configuración de cifrado fuerte, basado exclusivamente en TLS 1.2 o superior.

```
# Fragmento del fichero /etc/httpd/conf.d/ssl.conf
SSLProtocol -all +TLSv1.2
SSLHonorCipherOrder On
SSLCipherSuite
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:
ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS
```

2.5 Acerca del software de administración.

Para administrar la maquina se deben evitar protocolos inseguros (ej: no usar *telnet ni ftp y* emplear en su lugar *ssh/scp/sftp*) y se debe restringir el acceso a los servicios de administración que en ningún caso deben estar accesibles desde toda Internet.

De esta forma, servicios como *phpmyadmin*, o *webmin* deben estar siempre protegidos por usuario y contraseña, emplear cifrado https y en cualquier caso, nunca visibles directamente desde Internet.

También se debe evitar el uso de protocolos de administración gráfica no cifrados como *vnc* y emplear en su lugar protocolos seguros como *FreeNX* o *x2go*.

2.6 Copias consistentes de bases de datos.

De las máquinas virtuales IaaS se realiza una copia de seguridad diaria del sistema de ficheros, pero dicha copia de seguridad ignora los ficheros de motores de bases de datos. La razón es que una base de datos requiere de procedimientos propios para realizar copias de seguridad consistentes.

Por ejemplo, en el caso de una base de datos mysql proponemos el uso del siguiente script **/usr/local/bin/backup_mysql.sh** :

```
#!/bin/bash

FECHA=`date +%Y%m%d`

/usr/bin/mysqldump -E --single-transaction --all-databases -uroot "-
pCLAVE_ROOT_MYSQL"  > /var/backup/dump_mysql_${FECHA}.sql
/usr/bin/xz -9 /var/backup/dump_mysql_${FECHA}.sql
```

Este script debe lanzarse diariamente vía *cron, por ejemplo*:

```
#
# Volcado diario de la BBDD
#
28 23 1-31/1 * *       /usr/local/bin/backup_mysql.sh
```

Hay que tener en cuenta lo siguiente:

- El directorio **/var/backup** debe existir y debe tener los permisos adecuados para que únicamente el usuario que haga la copia de seguridad tenga acceso.
- El script **/usr/local/bin/backup_mysql.sh** únicamente debe tener permisos de lectura y ejecución para el usuario que realice la copia de seguridad.
- Es conveniente borrar automáticamente los ficheros de **/var/backup/** que superen cierta antigüedad (ej: 30 días) para no usar más espacio del necesario.

Es recomendable definir procedimientos similares de volcado diario para otros

motores de bases de datos, como *MongoDB*, *Postgresql* u *Oracle*.

De esta forma, en **/var/backup/** tendremos un volcado diario de nuestras bases de datos de forma comprimida. El backup diario sí realizará copia de estos ficheros de volcado, por lo que tendremos copias de seguridad consistentes de nuestras bases de datos.

2.7 Instalación de software adicional.

Siempre que sea posible se debe usar el sistema de control de paquetes **yum** para instalar nuevo software, procurando evitar instalar paquetes por otros medios (ej: a partir de un fichero *.tar.gz*). De esta forma, nos será mucho más sencillo mantener actualizado nuestro sistema.

A veces necesitamos versiones muy recientes de ciertos paquetes de software libre (ej: *mysql* o *php*), y estas versiones tan recientes no están disponibles en Red Hat / CentOS /Rocky Linux. En el caso de que los repositorios oficiales de Red Hat / CentOS / Rocky Linux no contengan el software libre que necesitamos, recomendamos el uso de estos repositorios adicionales, en este orden:

- *EPEL: https://fedoraproject.org/wiki/EPEL/es*
- *Repoforge: http://repoforge.org/*
- *IUS: https://iuscommunity.org/pages/GettingStarted.html*
- *Remi: http://blog.famillecollet.com/pages/Config-en*

Desafortunadamente, la gran mayoría del software comercial (ej: Mathematica) no se distribuye en formato paquetes RPM, por lo que en estos casos no hay más remedio que hacer instalaciones ad-hoc.

Por tanto, en resumen, se debe evitar en lo posible el uso de software no integrado con el sistema de gestión de paquetes.

2.8 Usuarios, permisos y contraseñas.

Los sistemas GNU/Linux heredan la filosofía de gestión de usuarios y permisos de UNIX. Por desgracia, es muy común que en las máquinas IaaS se emplee el usuario administrador para todas las tareas, siendo esto un error común a evitar.

La cuenta de administrador debe emplease únicamente cuando sea estrictamente necesario: instalación de nuevo software, parada y arranque de servicios, etc. Para el resto de tareas se debería emplear un usuario no privilegiado. En el caso de que varios usuarios compartan la máquina, es posible combinar usuarios, grupos y permisos para cubrir las necesidades de acceso de cada usuario/colectivo.

Respecto a los permisos de directorios y ficheros, debe regir el principio de mínimo privilegio: es un gran error emplear permisos laxos, y bajo ningún concepto se debe entender que aplicar permisos totales (777, rwx rwx rwx) sobre una carpeta o fichero es una medida adecuada para la prestación de un servicio.

La política de contraseñas es también crucial. Si el administrador o cualquier otro usuario de la máquina usa contraseñas débiles y fácilmente predecibles, tarde o temprano esa contraseña será averiguada. Una buena política de contraseñas podría ser la siguiente:

- *La **validez** de las contraseñas **es de un año**. Se configurarán avisos un mes antes de que caduquen para recordar el cambio.*
- *La **longitud** de la contraseña debe ser como **mínimo de 12 caracteres.***
- *La contraseña debe contener **al menos 4 caracteres alfabéticos,** de los cuales serán, al menos, dos letras mayúsculas y dos minúsculas.*
- *La contraseña debe contener **al menos 2 caracteres numéricos.***
- *La contraseña debe contener **al menos 2 caracteres especiales.***
- *El número máximo de repeticiones de caracteres adyacentes de la nueva contraseña es 4.*
- *La nueva contraseña no podrá contener el nombre de usuario o parte de él. El número máximo de caracteres en cadena coincidentes con el nombre de usuario es 3.*
- *Se deben evitar secuencias y repeticiones alfabéticas (como abcde o asdfg) y/o numéricas (como 12345 o 0000)*

- *La contraseña **no deberá** contener el **nombre o apellido** del usuario, **ni el documento** de identidad del mismo o su UVUS.*
- *La contraseña **no** debe contener palabras que aparezcan en un diccionario, como **días de la semana o nombres del mes**.*
- *No se deben utilizar las últimas tres contraseñas empleadas con anterioridad.*

Por último, se recomienda emplear el sistema *Security Enhanced Linux* (*SELinux),* integrado tanto en RedHat como en Rocky Linux para aumentar la seguridad del sistema.

2.9 Registros, logs y auditorías.

Los registros de logs suelen ser los grandes olvidados. Diariamente se generan automáticamente multitud de registros y alertas acerca del funcionamiento del sistema. Si estos logs no son revisados de forma periódica dejan de tener utilidad.

La primera sugerencia que realizamos es revisar diariamente el log de los sistemas. Si no se cuenta con tiempo para ello, recomendamos el uso de la herramienta *logwatch* que envía a diario un correo electrónico con un resumen de los hechos relevantes encontrados en los ficheros de logs generados el día anterior.

Otra sugerencia es centralizar todos los logs del sistema operativo en un único fichero[2], para mayor comodidad. Así, modificaríamos el fichero **/etc/rsyslog.conf** para incluir esta línea:

```
# Todos los logs de sistema se vuelcan de forma asíncrona en el fichero messages
*.*                                             -/var/log/messages
```

Es necesario relanzar el servicio **rsyslog** para que surta efecto el cambio.

Fragmento del fichero **/etc/logrotate/logrotate.conf**

```
# generar un fichero de log al día
daily

# dos años de logs
rotate 730

# crear ficheros vacíos tras el rotado del log
create
```

2 Esta línea se aplica exclusivamente a los logs generados por el sistema operativo. Otras aplicaciones, como apache o mysql generan sus propios logs en ficheros separados.

```
# usar la fecha como sufijo de rotado
dateext

# compresión de ficheros rotados

compresscmd /usr/bin/xz
uncompresscmd /usr/bin/unxz
compressext .xz
compressoptions -9
compress
```

2.10. Recursos adicionales.

Guía de seguridad oficial de Red Hat Enterprise Linux 7 (en ingles):

https://hdvirtual.us.es/discovirt/public.php?service=files&t=e605b454bb2e094454258fd51c223aa3

Guía oficial de seguridad de RHEL8:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/