Buenas prácticas para asegurar un servidor 

Linux es considerado un sistema operativo seguro. Sin embargo, se ha admitido que Linux está siendo utilizado más y más. Por ello, es también más atractivo para los hackers y crackers. Existen, algunos exploits que han sido descubiertos, han habido anuncios de seguridad que se han hecho públicos rápidamente y han sido reparados. Por lo tanto, aunque instale su servidor en Linux debe tomar medidas para mantener su servidor seguro. Pues muchas de las vulnerabilidades en nuestros sistemas no son inherentes únicamente al sistema operativo, sino a su configuración, uso y actualización.

Existen algunas medidas que se pueden tomar para mejorar en gran medida la segura de Linux. Aquí tienes las más comunes y utilizadas de ellas:

- **Medios de instalación:** Este elemento afectará a la velocidad de instalación y recuperación de un equipo. También influirá en la seguridad. Existen varios métodos:

  FTP – rápida, requiere una tarjeta de red, y un servidor de ftp preferiblemente conocido (como el ftp de la ULA)

  HTTP – también rápida, y algo más segura que hacer una llamada a un FTP público desconocido

  Samba – rápida, un buen método si dispones de una máquina windows (comparte el cdrom).

  NFS – no tan rápida, pero dado que nfs está implementado en la mayoría de las redes UNIX existentes (y NT tiene un servidor NFS de MS gratis), es casi indolora.

  CDROM – si tienes un lector de cdrom rápido, la mejor apuesta es introducir el cd y arrancar desde él, pulsar enter unas cuantas veces y ya estás listo. Tienes que estar atento del nivel de actualización del sistema, de ser necesario, debes realizar una actualización al finalizar la instalación.

  Disco duro – generalmente la más dolorosa, las ventanas confunden los nombres de fichero, la instalación desde una partición ext2 suele ser algo menos dolorosa.

  Imágenes ISO en CD - Si quieres tostar tu propia distribución X en CD, como el caso de ULAnix, este tipo de instalación permiten estandarizar los equipos que instalas y así facilitar su mantenimiento en el tiempo.

- **Actualiza las herramientas del sistema, las aplicaciones y el kernel:** La causa más común de ataques en un sistema es la inhabilidad de los administradores de mantener sus servidores al día con el proceso de actualizaciones. Mantener un

esquema de actualización regular del kernel, las herramientas y las utilidades te asegurará que tu sistema no está expuesto a los atacantes que conocen vulnerabilidades y exploits que ya están a su disposición. Para mantener un servidor linux al día puede conseguir más información aquí: actualizaciones en linux

- **Utiliza contraseñas sombra ó shadow password:** es altamente recomendable la utilización de este tipo de contraseñas, es una vulnerabilidad conocida del sistema operativo Unix y consiste en que el archivo de usuarios /etc/passwd tiene permisos de lectura para cualquier usuario y sólo de escritura para root. En este archivo hay un campo con el hash del password de cada usuario. Esta información puede ser utilizada por un atacante para descifrar una contraseña por medio de un ataque de fuerza bruta. Para evitar esta vulnerabilidad se crearon las contraseña sombra, que consiste en colocar el hash de la contraseña en un archivo /etc/shadow ó /etc/master.passwd en el que sólo root tiene permisos de lectura. Para más detalles de cómo utilizar este tipo de contraseñas puedes conseguir más información en: contraseñas en linux
- **Política de acceso:** Asegúrese de mantener una política de acceso y usuarios segura. Especialmente, para los usuarios con acceso al shell. Sus contraseñas deben ser complejas y cambiadas regularmente (por lo menos trimestralmente). Si administras más de un servidor, resiste la tentación de colocarles la misma contraseña (si un intruso lograr entrar a uno de tus servidores, podrá hacerlo a todos). Crea usuarios no root para las tareas no root, una práctica común entre los administradores es utilizar root para todas sus tareas, crea usuarios por cada persona que entrará al sistema y otorga permisos de acuerdo a las tareas que realizarán en el equipo. Utiliza sudo para proporcionar acceso a comandos privilegiados cuando sea necesario. Estas políticas te permitirán llevar un mejor control del uso del equipo y determinar un comportamiento extraño de los usuarios, incluyendo root.

  Utiliza las claves de las BIOS para mantener a los usuarios alejados de la BIOS (nunca deberían estar ahí, recuerda también que las BIOS viejas tienen claves universales.).

  Coloca clave al prompt de LILO.
- **Configura el servidor para que arranque únicamente del disco duro adecuado**.
- **Utiliza el shell seguro (ssh):** cambia el uso de acceso de telnet a ssh. Telnet es inseguro por 2 razones: Primero, sus sesiones no están cifradas, lo que implica, que todo lo que se transmite, incluyendo tu usuario y contraseña se transmiten en texto claro. Segundo, un puerto abierto de telnet es uno de los primero lugares que un craker buscará para tratar de conectarse a un servidor.

SSH provee un servicio de comunicación cifrado y comprimido, lo que mejora substancialmente la seguridad con respecto a la conexiones con telnet. Puedes proveer un tu servidor un servicio ssh como servidor y como cliente para permitir conexiones entrantes y salientes.

- **Utiliza cortafuegos:** para que incluso si instalan servicios estos no sean accesibles al resto del mundo. Para ello utiliza herramientas cómo iptables que permiten filtrado de paquetes de acuerdo a su protocolo, dirección ip, red, etc.
- **Restringe el acceso a servicios externos:** un error común en la configuración de un servidor es dejar abierto el uso a servicios externos que no serán utilizados y algunos que son inseguros, como telnet. ç

  Para ello debes editar el archivo “etc/hosts.allow” y el archivo “`/etc/hosts.deny`" para restringir el acceso a tus servicios. En el siguiente ejemplo puedes ver como se restringe el acceso solo al servicio ssh, negando las conexiones por telnet.

  Primero, en el archivo “/etc/hosts.allow”:

  ```
  # vi /etc/hosts.allow
  ```
  sshd: 150.185.180.0/255.255.254.0

  Segundo, niega el acceso al resto de los servicios escribiendo en el /etc/hosts.deny

  # vi /etc/hosts.deny

  ALL:ALL except 127.0.0.1: Deny

- **Apagar y desinstalar servicios innecesarios:** Para conocer los servicios que se están utlizando y qué puertos están abiertos en tu servidor ejecuta los siguientes comandos:

  # nmap localhost (para conocer los servicios abiertos)

  Starting Nmap 4.20 (http://insecure.org) at 2007-08-08 14:30 VET

  Warning: Hostname localhost resolves to 2 IPs. Using 192.168.2.1.

  Interesting ports on sun-ray-eth1 (192.168.2.1):

  Not shown: 1691 closed ports

  PORT STATE SERVICE

  22/tcp open ssh

  25/tcp open smtp

  80/tcp open http

Nmap finished: 1 IP address (1 host up) scanned in 0.223 seconds

Si existe algún servicio innecesario puedes deshabilitarlo colocando un el carácter de comentario “#” en el servicio que no necesitas en el archivo ``/etc/inetd.conf". Después de realizar los cambios, entrando como root, re-inicia el demonio inetd (utilizando el script “/etc/rc.d/init.d/inet restart" para que tome los cambios que realizaste.

 # runlevel (para conocer los servicios que se están ejecutando)

 # ls /etc/rc2.d/

Para impedir que los servicios que no se están utilizando se ejecuten utilice el comando update-rc.d .Una vez que se impida el arranque de estos servicios hay que detenerlos.

Por ejemplo para detener el servicio apache:

 # update-rc.d -f apache remove

update-rc.d: /etc/init.d/apache exists during rc.d purge (continuing)

Removing any system startup links for /etc/init.d/apache ...

/etc/rc0.d/K20apache

/etc/rc1.d/K20apache

/etc/rc2.d/S20apache

/etc/rc3.d/S20apache

/etc/rc4.d/S20apache

/etc/rc5.d/S20apache

/etc/rc6.d/K20apache

 # /etc/init.d/apache stop

Stopping web server: apache.

Una vez realizados los cambios reinicia el servidor y verifica que los puertos para los servicios innecesarios están cerrados y que no se están ejecutando.

- **Evite el uso de los comandos “r” (Ej. rlogin, rsh.)**

En el uso diario de Linux es frecuente la ejecución de comandos en máquinas remotas. Tradicionalmente esta necesidad se resolvía con los comandos "r" esto es: **rlogin**, **rsh** y **rcp**. Estos comandos lanzan un shell en la máquina remota y permiten al usuario ejecutar comandos. El usuario debe usar una usuario en la máquina remota, por lo que debe pasar por los métodos de autenticación. Los comandos r usan la autenticación simple de usuario y contraseña, y utilizan una conexión en texto claro, por lo que estas pueden ser interceptadas por la red.

La autenticación para los comandos "r" también se puede controlar desde algunos

archivos de configuración como son:

- `/etc/hosts.equiv` : a nivel de sistema, equivalencia entre usuarios de distintas máquinas. Se puede evitar la introducción de contraseñas.
- `$HOME/.rhosts`: a nivel de usuario, permite el acceso a usuarios de otras máquinas sin utilizar contraseña.

Cómo deshabilitar los comandos "r"

Los comandos "r" se controlan desde el demonio inet, para deshabilitarlos puedes comentar sus entradas en el fichero /etc/inetd.conf. Si tu distribución utiliza Xinetd puedes comprobar si están deshabilitados con el siguiente comando:

root@localhost root]# chkconfig --list|grep -E "rlogin|shell|exec"

kshell: off

rexec: off

rlogin: off

En nuestro caso ya están deshabilitados. Si alguno de ellos no lo estuviera puedes conseguirlo con el comando:

root@localhost root]# chkconfig rlogin off

fuente: Enrique Galdú. Ssh sustituye a rsh Después de la r viene la s.

- **Revise los permisos y la propiedad de los archivos de configuración del sistema y los servicios.**

/etc/passwd

El archivo de contraseñas es sin discusión el fichero más crítico en Linux (y en la mayoría de otros Unix). Contiene el mapa de nombres de usuarios, identificaciones de usuarios y la ID del grupo primario al que pertenece esa persona. Cómo se menciono en un punto anterior es nuestra recomendación que en se utilice las contraseñas shadow ya que este archivo TIENE que ser legible por todo el mundo para que puedan funcionar comandos básicos del sistema. Sólo el usuario root debe tener permisos de escritura y ejecución sobre él.

/etc/shadow

El archivo de shadow alberga pares de nombres de usuario y contraseñas, así como información contable, como la fecha de expiración, y otros campos especiales. Este archivo debe protegerse a toda costa, y sólo el usuario root debe tener acceso de lectura y escritura a él.

/etc/groups

El archivo de grupos contiene toda la información de pertenencia a grupos, y

opcionalmente elementos como la contraseña del grupo (generalmente almacenado en gshadow en los sistemas actuales), este fichero debe ser legible por el mundo para que el sistema funcione correctamente, su formato es:

nombregrupo:contraseña_cifrada:GID:miembro1,miembro2,miembro3

/etc/gshadow

Similar al fichero shadow de contraseñas, este fichero contiene los grupos, contraseñas y miembros. De nuevo, este fichero debería ser protegido a toda costa, y sólo el usuario root debería tener permiso de lectura al mismo.

/etc/login.defs

Este fichero (`/etc/login.defs`) te permite definir algunos valores por defecto para diferentes programas como useradd y expiración de contraseñas. Tiende a variar ligeramente entre distribuciones e incluso entre versiones, pero suele estar bien comentado y tiende a contener los valores por defecto.

/etc/shells

El fichero de shells contiene una lista de shells válidos, si el shell por omisión de un usuario no aparece listado aquí, no podrá hacer login interactivamente.

/etc/securetty

Este fichero contiene una lista de tty's desde los que el root puede hacer un login. Los tty's de la consola suelen ir de `/dev/tty1` a `/dev/tty6`. Los puertos serie (pongamos que quieres hacer login como root desde módem) son `/dev/ttyS0` y superiores por lo general. Si quieres permitirle al root hacer login vía red (una muy mala idea, utiliza sudo) entonces añade `/dev/ttyp1` y superiores (si hay 30 usuarios conectados y el root intenta conectar, el root aparecerá como procedente de `/dev/ttyp31`). Generalmente, sólo se debería permitir conectar al root desde `/dev/tty1`, y es aconsejable desabilitar la cuenta de root, sin embargo antes de hacer esto, por favor, instala sudo o un programa que permita al root acceder a comandos.

- **Realice auditorías:** Este alerta, ejecute auditorías aleatorias a distintos aspectos de su servidor y sus servicios. Pueden hacerse cosas tan simples como revisar los archivos de sistema, por ejemplo: el auth.log y revise por entradas de usuarios sospechosos, IP desconocidas, etc. Reporte cualquier caso de ataque o intento de ataque a nuestro correo electrónico gsc@ula.ve

  Mantenga esquemas de revisión de:

  (a) Accesos y permisos a los archivos y directorios

  (b) Las contraseñas de sus usuarios

  (c) Accesos de los usuarios

(d) Observa regularmente la tabla de procesos, puertos abiertos, software instalado, etc. en busca de cambios inesperados.

Para realizar un revisión de los distintos archivos de registro del sistema de una forma fácil de visualizar utiliza herramientas como logwatch con el comando:

 # logwatch --range all --print | more

Si desea revisar la fortaleza de las contraseñas utilizadas por sus usuarios utiliza herramientas con John the Riper y CrackLib que te permitirán definir las contraseñas débiles en tu sistema para así solicitar a sus usuarios que la cambien por contraseñas con mayor complejidad.

- **Mantenga un esquema de revisión de los archivos de sistema o aplicaciones de detección y prevención de intrusos:** Considera la instalación de programas como “Tripwire” (véase, http://www.tripwiresecurity.com/) para detectar intrusos y “Abacus Sentry” (véase http://www.psionic.com/abacus/) con el que puedes prevenirlos. Los archivos de sistema pueden suministrar información importante sobre las posibles intrusiones o intentos de intrusión en su servidor, configure un reporte por correo electrónico de la actividad diaria de se equipo, puedes usar logwatch (http://www2.logwatch.org) ó sar (system activity report).

Actualizaciones en Linux

Actualización del kernel

Es necesario regularmente actualizar el kernel de Linux. Esto te permitirá tener acceso a las nuevas funcionalidad del sistema y reparar cualquier vulnerabilidad que se haya encontrado y reparado en versiones anteriores.

Muchas veces el kernel no se actualiza por el mito de la dificultad y riesgo que ello representa. Sin embargo, permanecer con kernels desactualizados representa un riesgo mayor.

Puedes encontrar noticias o anuncios de nuevas versiones a través de múltiples fuentes de información sobre tu instalación de linux. Aquí hay algunas de ellas: comp.os.linux.announce, http://freshmeat.net/ ó http://slashdot.org/

Al momento de actualizarte es importante tomar en cuenta qué tipo de equipo estás utilizando, para escoger qué tipo de actualización hacer: estable, pruebas o desarrollo. Si tu equipo es crítico para la misión de tu empresa, no es recomendable utilizar las versiones de desarrollo o prueba. Estas versiones traen nuevos servicios y utilidades, pero no se consideran a prueba de fallas.

Contraseñas en Linux y el formato de archivo sombra (shadow file format)

Los sistemas tradicionales de Unix mantienen la información de las cuentas de sus usuarios en un archivo en “/etc/passwd”, aquí se incluye la contraseña cifrada de cada usuario. Como este archivo es utilizado por múltiples aplicaciones que necesitan utilizar la relación usuario-contraseña necesita otorgarse permisos de lecturas a todo el mundo. Este hecho representa una vulnerabilidad del sistema.

Otra forma de guardar la información de las cuentas de usuarios es el formato de contraseña sombra (shadow password). Igual que el método tradicional se guarda la información en el “/etc/passwd”. Sin embargo, la contraseña es guardada con un sólo carácter “x”. En un segundo archivo llamado “/etc/shadow” se guarda la contraseña cifrada y otra información del usuario como la expiración de la cuentas, etc. Este archivo “/etc/shadow” sólo puede ser leída por el usuario root. Eliminando así el riesgo de que otro usuario pueda leerla y obtener por medio de un ataque de fuerza bruta la contraseña del usuario.

Muchas distribuciones de Linux en la actualidad te obligan a utilizar la versión de la contraseña sombra. Sin embargo, es nuestra recomendación que antes de decidir qué tipo de instalación se va a utilizar, revisar si este formato de contraseña ya viene por omisión en

el paquete.

Términos y expresiones del filtrado de paquetes

fuente: 2006. Andreasson Oskar. Iptables Tutorial 1.2.2.

http://iptables-tutorial.frozentux.net/iptables-tutorial.html

A continuación encontrará los términos más comunes utilizados en el filtrado de paquetes que le permitirá comprender mejor los detalles de los comandos para la configuración de iptables:

- Drop/Deny (soltar/denegar) – Cuando un paquete se suelta o deniega simplimente es borrado y no se toman otras acciones. No habrá respuesta al host de que el paquete fue denegado. El paquete simplemente desaparece.
- Reject (rechazar) – Este término se utiliza de igual forma que el drop/deny para negar paquetes que cumplan con una regla o política, excepto que se envía una respuesta al emisor del paquete indicandole que el mismo ha sido rechazdo. Esta respuesta puede ser específica o calculada automáticamente según variables indicadas con anterioridad (hasta la fecha con iptables está funcionalidad no está disponible por loq que no se puede especificar en las razones por las que se nego el paquete). Está habilidad puede ser útil en algunas circunstancias cómo ataques de denegación de servicio)
- State (estado) – Se especifica el estado de un paquete en lugar de un conjunto de paquetes. Por ejemplo, si el primer paquete que recibe el firewall reconoce un paquete nuevo (el paquete SYN en una conección TCP) o si es parte de una conexón establecida que el firewall ya conoce.
- Chain (cadena) – Una cadena contiene un conjunto de reglas que son aplicadas a los paquetes de una comunicación. Cada cadena tiene un propósito específico igual que área de aplicación (sólo paquetes destinados a un equipo).
- Table (tablas) – cada tablas tiene un propósito específico y en iptables son 4. Raw, nat, mangle y filter. Por ejemplo, la tabla filter está diseñada específicamente para el filtrado de paquetes, mientras que la nat para la traducción de los paquetes.
- Target (destino) – Por lo general una regla tiene un destino. Si la regla se cumple las especificaciones del target o destino le indican al firewall a dónde irá el paquete.
- Jump (salto) – La instrucción de salto está relacionada a la de destino. Un salto está escrito igual que una instrucción de destino en iptables, la diferencia que tienen que en lugar de escribir el nombre del destino se escribe la próxima cadena del mensaje. Si la regla se cumple, el paquete se enviará a esta segunda cadena y se procesará de la misma forma.

- Accept (aceptar) – para aceptar un paquete en un firewall se hace lo opuesto de la instrucción drop/deny o de reject y lo que se hace con el paquete es dejarlo pasar.

Referencias

http://labmice.techtarget.com/articles/securingwin2000.htm

Stearns William. Essential security checks for Linux Systems. 2003.

http://www.dragonjar.org/77-consejos-de-seguridad-informatica.xhtml

Enrique Galdú. Ssh sustituye a rsh Después de la r viene la s.

José F. Torres. Practicas básicas de de seguridad en Linux. 2da. Escuela Venezolana de Seguridad de Cómputo. Agosto 2006.

Universidad Autonoma de Madrid. Guía básica de seguridad para Windows NT. http://www.uam.es/servicios/ti/servicios/ss/rec/winnt.html