AndalucíaCERT
CENTRO DE SEGURIDAD TIC

Informe de divulgación
Bastionado de Sistemas (I)

Tipo de documento:	*Informe*
Autor del documento:	*AndalucíaCERT*
Código del Documento:	*CERT-IF-2045-220812*
Edición:	*0*
Categoría	*Público*
Fecha de elaboración:	*22/08/2012*
N° de Páginas	*1 de 17*

1 TABLA DE CONTENIDOS

2 OBJETO

El objeto de este documento es proporcionar al personal de la Junta de Andalucía una serie de puntos a tener en cuenta para fortificar los servidores de nuestro entorno y reducir las posibilidades de que nuestros sistemas se vean comprometidos.

3 ALCANCE

Este documento va destinado al personal de la Junta de Andalucía y al público en general. El tema que se aborda consta de un fondo principalmente técnico. Debido a la profundidad de la temática, el informe completo sobre bastionado de sistemas ha sido dividido en dos entregas. En la presente se pretende aportar una visión general y puntos relevantes del bastionado de sistemas, sin entrar en ninguna tecnología concreta.

4 INTRODUCCIÓN

> "*Haciéndole la vida difícil al atacante. Ese es el concepto que está detrás del Hardening de sistemas operativos. Hardening es una acción compuesta por un conjunto de actividades que son llevadas a cabo por el administrador de un sistema operativo para reforzar al máximo posible la seguridad de su equipo.*
>
> *Su propósito, entorpecer la labor del atacante y ganar tiempo para poder minimizar las consecuencias de un inminente incidente de seguridad e incluso, en algunos casos, evitar que éste se concrete en su totalidad."*
>
> *Luis Montenegro, Windows y Security MVP, 2007*

Es posible que alguna vez haya escuchado que "un sistema totalmente seguro es aquél que se encuentra desconectado y aislado". Como supondrá, esta situación no es realista. Para que una organización como la Junta de Andalucía funcione correctamente necesita establecer ciertos canales de comunicación y acceso entre sus empleados, clientes, proveedores, etc; los servicios que se ofrecen y los sistemas que los prestan.

Para establecer esos canales se necesitan abrir ciertas puertas en estos sistemas y servidores. Es lógico pensar que un intruso siempre intentará entrar a través de una puerta que está ya abierta, o al menos que se deja abrir fácilmente, antes que a través de la pared.

Asegurar que nuestros sistemas jamás serán accedidos de forma ilegítima es complicado. Sin embargo, si podemos concentrar esfuerzos que en sea bastante difícil acceder.

Bastionar un sistema consiste en reducir el área de superficie de ataque del mismo, eliminando tantos riesgos para la seguridad como sea posible. Este proceso también se conoce como "hardening" que podría traducirse como endurecimiento o fortificación.

Algunos beneficios que se puede obtener de invertir esfuerzos en bastionar un sistema son:

- Asegurar que los recursos críticos sean capaces de resistir a la explotación de vulnerabilidades conocidas.
- Contar con una configuración de referencia base segura, con la que llevar a cabo un despliegue rápido.
- Facilita la auditoría de un servidor frente a cambios inesperados.
- Mejora la seguridad de sus sistemas frente a amenazas internas y externas.
- Reduce los riesgos asociados con fraude y error humano.

Con este documento tratamos de proporcionar al lector una sencilla introducción a la fortificación de sistemas. Para ello se ha distribuido el documento teniendo en cuenta los principales puntos en los que debemos prestar especial atención a la hora de llevar a cabo esta labor:

- Aplicaciones y servicios
- Acceso remoto
- Acceso local
- Sistema operativo

5 APLICACIONES Y SERVICIOS

Es recomendable separar las funciones de los servidores de forma que cada servidor proporcione funciones muy concretos. Cuantas más tareas distintas realice un servidor, más aplicaciones tendrá que tener instaladas y mas servicios en ejecución, por lo que aumenta las posibilidades de encontrar una vulnerabilidad en alguna de éstas, aumentando consecuentemente las posibilidades de ser comprometido.

5.1 APLICACIONES

Actualmente, muchos sistemas operativos están diseñados para presentar al usuario el máximo número de opciones y aplicaciones activas posibles.

Todos los programas cuentan con debilidades que pueden ser aprovechadas por atacantes para comprometer el sistema en el que se encuentra instalado. Por tanto, cuantos mas programas haya instalado en el sistema, mayor cantidad de debilidades podrán ser encontradas y mayor riesgo para nuestro sistema ya que aumentarán los puntos que un atacante pueda aprovechar.

5.1.1 Instalar sólo el software necesario

Si hablamos de un equipo destinado a realizar funciones de puesto de trabajo de un usuario, es lógico que éste tenga instalado muchas y muy diversas aplicaciones. Sin embargo, si hablamos de un servidor, esto no debe ser lo común. Normalmente un servidor está destinado a cumplir una función mas específica que un equipo de usuario, y por tanto, no deberá tener instalado mas software que aquel que necesite para proporcionar las funciones para las que está destinado.

Por ejemplo, en general no hay necesidad de tener entorno gráfico en un servidor que sólo será accesible mediante consola.

Todo software que instalemos de más provocará la apertura de nuevas puertas de entrada a nuestro sistema para los atacantes.

Algunas aplicaciones comúnmente innecesarias en servidores son:

- Un gestor de ventanas. Las aplicaciones de servidor (WEB, FTP, servidores mail, etc...) no suelen requerir sistemas de gestión gráficos.
- Herramientas de ofimática (OpenOffice, MS Office, ...).
- Aplicaciones de escritorio en general (Navegador web, cliente de correo, reproductores multimedia, aplicaciones de diseño gráfico, etc).

- Programas asociados al desarrollo de aplicaciones en máquinas expuestas a Internet (compiladores, librerías de desarrollo, etc...).
- Herramientas de monitorización de red (wireshark, tcpdump, ngrep, etc...).

5.1.2 Mantener el software actualizado

Una vez determinado el software que debe permanecer en el sistema, éste deberá mantenerse actualizado. Las actualizaciones de los programas vienen justificadas principalmente por dos motivos:

- Corregir las vulnerabilidades conocidas.
- Proporcionar nuevas funcionalidades o mejoras respecto a las versiones anteriores.

La gran mayoría de sistemas operativos en la actualidad poseen mecanismos de actualización de software de manera automática. Sin embargo, en el caso de aplicaciones de servidores esto no es tan usual debido a que puede provocar incompatibilidades con las aplicaciones clientes.

Por esta razón es importante ser especialmente cuidadosos a la hora de añadir, eliminar o actualizar software en los **sistemas de producción**. En líneas generales se recomienda probar todos los cambios que vayamos a realizar en estos sistemas en un sistema de pre-producción de similares características y configuración.

En general, se debe contar con una **política de actualización** del software en los servidores.

Existen aplicaciones que detectan programas vulnerables y desactualizados que requieran de parches de seguridad, así como configuraciones incorrectas que pueden exponer tu equipo a ataques:

- Secunia PSI (Windows): http://secunia.com/products/consumer/psi/
- Inteco CONAN (Windows): https://conan.cert.inteco.es/analizador.php

5.1.3 Integridad del software

La comprobación de la integridad de un archivo o programa es importante pues nos permite averiguar si algún dato del mismo ha variado desde su creación. Hay múltiples motivos por los que un archivo puede verse modificado: un error, un corte en la transmisión del archivo o porque un atacante lo haya modificado para alterar su funcionamiento y que realice acciones maliciosas, como por ejemplo, abrir una puerta trasera (backdoor) en el sistema.

En entornos de código abierto es posible revisar y compilar a nuestro antojo el código completo de los programas, teniendo de esta manera un mayor control sobre el software que instalamos en nuestro sistema. Sin embargo en la práctica es algo tedioso, por lo que habitualmente se recurre a la descarga de

programas y paquetes ya compilados para nuestra plataforma. Por tanto, ¿cómo podemos estar seguros de que el programa descargado es realmente el generado por el autor legitimo del software?

Para la comprobación de la integridad de archivos habitualmente se usan **funciones hash criptográficas**. Éstas, mediante la aplicación de un algoritmo determinado, convierten un archivo de cualquier tamaño en una cadena de una longitud constante, lo que se denomina resumen criptográfico o "message digest". Así pues, a partir de un flujo de datos de un número indeterminado de bits es posible obtener un número constante que lo identifica de manera unívoca.

Los dos algoritmos más utilizados para la verificación de integridad son MD5 y SHA256. Para calcular el resumen de cualquier archivo podemos usar el programa **md5sum** que se encuentra disponible tanto en sistemas Linux como Windows. El resultado de esta operación debe estar también disponible en la página de descarga para comprobar que coinciden y que, por tanto, los archivos no han sido modificados:

$md5sum <archivo>
f01da331f7c1aa76c04552fb9f4b2c05 *<archivo>*

$sha256sum <archivo>
c53ff74c34fd6dfef1be1cff51bb8ad11097ce9073c6c2b0daad2903cc61a106 *<archivo>*

Otra opción mas sofisticada, pero que evidentemente consume mas recursos, es usar los conocidos monitores de integridad. Éstos se encargan de monitorizar y alertar de cambios específicos de ficheros en un rango de sistemas. Los programas de comprobación de integridad más usados son Tripwire, AIDE y Samhain.

5.2 SERVICIOS

Al igual que con los programas instalados, hay que prestar atención a los servicios que se ejecutan en el sistema, y determinar cuales son aquellos que son realmente necesarios y cuales no lo son. Un servicio en ejecución implica un puerto abierto en el servidor, y por tanto una entrada para un atacante.

Aunque la máxima es dejar tan pocos servicios ejecutándose en el sistema como sea posible, obviamente una máquina debe de tener los servicios que necesite disponibles para cumplir con sus funciones. No debemos olvidar que uno de los objetivos de la seguridad en un sistema es preservar la disponibilidad de los servicios que se ejecutan.

En sistemas Linux podemos comprobar los servicios que están a la escucha en el servidor mediante el comando **netstat -putan**. En Windows sería con **netstat -a -b -n**.

5.2.1 Servicios potencialmente inseguros

En potencia, cualquier servicio de red es inseguro. Sin embargo, existen algunos de ellos que por su implementación son clasificados como inseguros. Estos servicios deben ser deshabilitados en caso de no ser necesarios, o debidamente protegidos tras un cortafuegos en caso de que se necesiten.

Típicamente un servicio se considera inseguro cuando:

- Transmite datos confidenciales sin cifrar.
- Tiene vulnerabilidades conocidas.

RPC (135/tcp): Protocolo para llamada a procedimientos remotos. Permite que dos procesos, ubicados en la misma máquina o en diferentes máquinas conectadas, se comuniquen entre si. Habilita el intercambio de datos y permite solicitar funcionalidades residentes en otros procesos.

Este tipo servicios no se consideran muy seguros y normalmente no deben ser usados en servidores en producción, por lo que se recomienda que sean **deshabilitados** siempre que sea posible.

Servicios r-: Permiten abrir sesiones remotas y transferir archivos utilizando como autenticación algunas combinaciones de usuario/contraseña y dirección IP de origen. Los datos son transmitidos en claro y los orígenes de las direcciones IP pueden ser falseadas.

Estos servicios son los siguientes:

- rlogin (531/tcp): Terminal virtual en sistemas Unix. Similar a Telnet.
- rsh (514/tcp): Ejecución de comandos en una máquina remota sin necesidad de acceder a ella.
- rcp (vía rsh): Copiado de ficheros entre diferentes máquinas.

Se recomienda el uso de SSH como sustituto de este tipo de servicios.

Servicios de correo (sendmail, qmail, postfix, etc..): Aunque muchas personas piensan que este servicio, habilitado por defecto en muchas versiones de Linux, debe ejecutarse continuamente como un demonio debido a que se utiliza para enviarse correos a si mismo (notificaciones administrativas, por ejemplo), esto no es así. **Sólo debería ejecutarse como demonio en servidores que deban recibir correos de otros sistemas**.

Telnet (23/tcp), FTP (21/tcp), POP (110/tcp): Estos tres protocolos tienen desafortunadamente una característica común: Los datos de autenticación son enviados en texto claro. Telnet y FTP son sustituibles fácilmente por SSH y sus utilidades de trasferencia de archivos SCP y SFTP. En el caso de POP, debe ser configurado a través de un protocolo seguro (SSL). Se recomienda su desinstalación del sistema si no se hace uso de ellos.

NFS y SMB: Transmiten la información sin cifrar. No son seguros por defecto y no deberían usarse en servidores críticos.

6 ACCESO REMOTO

Para proteger nuestros servidores de los accesos procedentes de otras redes se usan sistemas de protección perimetral como firewalls, proxys, etc. Sin embargo, pese a que son bastante fiables estas medidas de protección, siempre existirán puertas de acceso abiertas que se tendrán que dejar sin bloquear para que puedan acceder los administradores, para que algunos servicios se comuniquen con otras máquinas, o simplemente para que el servidor pueda proporcionar su función a los clientes. De hecho, es necesario asegurar que dichos puntos de entrada estén siempre disponibles y accesibles para aquellas partes legítimas.

Siguiendo el principio de acceso mas fácil, por regla general un atacante intentará acceder a nuestro servidor a través de esas puertas. Por tanto, en este apartado contemplaremos la fortificación de esos puntos de entrada, que no podemos deshabilitar ni filtrar por completo.

6.1 Aislar la máquina del resto de equipos de la misma red

Normalmente siempre suele existir un firewall a la entrada de la red de servidores, pero las reglas de los firewall no suelen contener ningún tipo de restricciones de conexión entre los servidores situados en la misma red. Si un servidor no debe tener ningún tipo de tráfico con otro servidor de esa red, se debe establecer un conjunto de reglas que no permitan que exista tráfico entre ambas máquinas. De no ser así, si un atacante consigue el control de una máquina de esa red, podrá atacar al resto de máquinas desde dicho servidor sin que el tráfico sea filtrado. A su vez, se debería emplear envolturas y reglas de filtrado a nivel del servidor (local) para que en caso de que el router fuera atacado, el servidor no estuviera desprotegido.

6.2 Administración remota

Lo mas extendido para la administración remota de servidores basados en Windows es el uso de **RDP** (Remote Desktop Protocol). Este protocolo permite cifrado de 128 bits utilizando el algoritmo criptográfico RC4. En sistemas Linux, usar **SSH** es la mejor forma de administrar todos los servidores desde un equipo.

Sin embargo, ninguno de estos protocolos son impenetrables. Los protocolos pueden ser vulnerados por parte de un atacante, tanto interno como externo a la red del servidor. Por ello se ofrecen algunas recomendaciones para mejorar la seguridad de los servidores que administremos, bien vía SSH o vía RDP:

- Utilizar la versión mas segura del protocolo.
- Aplicar los últimos parches de seguridad (en el caso de RDP recordar Microsoft MS12-020).
- Deshabilitar o renombrar el usuario root/administrador.
- Limitar el acceso solo a cuentas autorizadas.
- Utilizar un puerto distinto al establecido por defecto:

  - En SSH usar un puerto distinto al 22/tcp.
  - En RDP usar un puerto distinto al 3389/tcp.
- Abrir el puerto del servicio solo a IPs autorizadas .
- En SSH, deshabilitar el uso de contraseñas.
  - Utilizar llaves criptográficas (DSA / RSA) .
- En RDP usar un nivel alto de encriptación (128-bits).
- Limitar el número de intentos de acceso fallidos en un corto periodo de tiempo.

Por otro lado, no se recomienda permitir la administración vía SSH desde máquinas situadas en una red no confiable. En estos casos es mejor conectar mediante túneles VPN.

7 ACCESO LOCAL

Es habitual que durante el bastionado de un servidor se preste gran atención a protegerlo contra atacantes externos o remotos. Pero, si un atacante es capaz de burlar a los sistemas de protección perimetral, y consigue conexión directa con la máquina víctima, todavía no tiene que estar todo perdido si hemos reforzado la seguridad local del sistema adecuadamente.

Es muy importante no obviar la seguridad interna del sistema. Ex-empleados molestos que conservan credenciales de acceso, ingeniería social a personal interno, errores humanos, son solo algunos ejemplos. El impacto que puede producir un ataque accidental o intencionado dependerá en gran medida de las medidas de seguridad aplicadas a proteger el acceso local de nuestro sistema.

Nota: En esta sección no entraremos a aquellos puntos referidos a la seguridad física.

7.1 Cuentas de usuario

En el ámbito de la seguridad existe un principio básico que se ha de aplicar a todo proceso: **el principio de mínimo privilegio**. Se trata de una de las piedras angulares de la seguridad: realizar las tareas necesarias con los mínimos privilegios; así cualquier fallo, accidente o vulnerabilidad tiene también un impacto reducido.

En base a esto, debemos tener bien configuradas las cuentas de usuario para que dispongan de los privilegias necesarios (ni mas, ni menos) para realizar las tareas que necesita. Podemos diferenciar varios tipos de cuentas de usuario:

- **Administrador**: Permite tener el control total del equipo, y por tanto realizar todo tipo de actividades como instalar y desinstalar programas, agregar un nuevo componente de hardware, configurar la red, etc. Todo sistema debe tener una cuenta de administrador, que será la que utilicemos para habilitar el resto de cuentas de usuario. Todas las acciones que realicemos con la cuenta de administrador pueden repercutir al resto de cuentas.

- **Usuario limitado**: Las acciones realizadas en esta cuenta no afectan al resto de cuentas ni al correcto funcionamiento del sistema.

- **Aplicación**: Son cuentas asignadas a las aplicaciones para que sean las que usen éstas durante su funcionamiento en el sistema. No suelen tener acceso a entornos de usuario como consola de comandos, entorno gráfico, etc. Solo tienen acceso a los recursos que necesita la aplicación.

- **Invitado**: Similar a la limitada. Está pensada para que usuarios que no tienen una cuenta propia en el equipo puedan utilizarla en un momento puntual. En la medida de lo posible no se recomienda su uso.

En el caso de servidores, se recomienda no tener ninguna cuenta de invitado, y el mínimo de cuentas de administrador posibles. En cuanto a las cuentas de aplicación, serán necesarias tantas como aplicaciones las necesiten, y usuarios limitados, una por cada usuario que acceda.

Se debe **evitar el uso de cuentas genéricas** para los usuarios, y **en especial para administradores**. Lo recomendable es que **cada usuario tenga una cuenta nominal**.

Las cuentas de administrador no están pensadas para utilizarlas en el día a día, suponiendo un riesgo para la seguridad hacer de ellas un uso cotidiano. En el uso diario se recomienda utilizar, para cada uno de los usuarios del sistema, una cuenta con privilegios limitados con las que poder desempeñar las actividades cotidianas.

Para más información sobre cómo gestionar las cuentas de usuario consulta la documentación oficial.

- Microsoft Windows:
  - Guía para gestionar las cuentas de usuario en Windows XP
  - Guía para gestionar las cuentas de usuario en Windows Vista
  - Guía para gestionar las cuentas de usuario en Windows 7
- Linux Ubuntu:
  - Guía para gestionar las cuentas de usuario en Linux Ubuntu .

7.2 Política de contraseñas

Por mucho que se proteja un sistema, algunos usuarios son descuidados y no se preocupan lo suficiente a la hora de asignar una contraseña, o a la hora de conservarla. Cuando eso ocurre, un potencial atacante que ya hubiese superado la primera barrera, la remota, tendría acceso al sistema gracias una cuenta débil proporcionada por un usuario legítimo. Si además de eso, el usuario cuenta con muchos privilegios, la cosa se torna a peor.

Para evitar esta clase de problemas es interesante obligar a los usuarios a cumplir una serie de requisitos en lo que respecta a la definición y actualización de sus contraseñas:

- Determinar vigencia máxima de la contraseña, y por tanto, obligar a redefinir su contraseña cada cierto periodo de tiempo. También es posible determinar la vigencia mínima de la contraseña con el fin de que no se cambien con demasiada frecuencia (mas de 1 o 2 veces al día).

- Establecer un historial de contraseñas para que se recuerden varias contraseñas ya usadas anteriormente. Con esta configuración se evita que los usuarios usen la misma contraseña cuando ésta caduca.

- Definir la longitud mínima de la contraseña de forma que deban estar formadas por un número mínimo de caracteres especificado. Con esta configuración evitamos, por ejemplo, que los usuarios puedan utilizar contraseñas en blanco. A partir de 8 caracteres podría considerarse una longitud adecuada.

- Determinar requerimientos de complejidad. Por ejemplo, podemos imponer la restricción de que la contraseña esté compuesta obligatoriamente de al menos una mayúscula, un dígito y un carácter especial, y que no contenga ninguna porción del nombre de usuario ni los apellidos del mismo.

7.3 Límite de recursos

Con el fin de evitar que desde la cuenta de un usuario se pueda volver inestable el sistema, bien por accidente, o bien porque esa cuenta sea comprometida y se use para realizar ciertos ataques, es importante establecer determinados límites en cuanto a la cantidad de recursos que puede manejar un usuario en el servidor.

Algunos recursos que podemos limitar son los siguiente:

- **Cuotas de disco**: Controlando el máximo de espacio que se puede usar evitamos que un usuario colapse todo el espacio libre de forma accidental o intencionada.

- **Uso de CPU**: Máxima cantidad de tiempo de CPU que los procesos de un usuario pueden consumir.

- **Memoria**: Máxima cantidad de memoria RAM que un proceso puede utilizar.

- **Procesos**: Número máximo de procesos que un usuario puede iniciar. Si obviamos este paso seremos vulnerables, por ejemplo, a una bomba fork, la cual puede causar una denegación de servicio.

- **Tamaño de ficheros**: Tamaño máximo que puede llegar a tener cada fichero de forma individual del usuario.

8 SISTEMA OPERATIVO

8.1 Actualizaciones

Es fundamental que nuestros sistemas se encuentre actualizados y con los últimos parches de seguridad instalado. Para ello es imprescindible contar con una **política de actualización** de los sistemas desplegados.

La mayoría de los ataques actuales se basan en explotar vulnerabilidades conocidas de un servicio o aplicación instalada en el sistema. Lo mismo ocurre con el malware. Cada nuevo virus, troyano o gusano basa su infección en aprovecharse de vulnerabilidades conocidas.

8.2 Procesos de arranque

El boot o secuencia de arranque es realizado por una serie de procesos que pueden ser objeto de ataque y que debemos securizar. El boot proporciona una interfaz de comandos más del sistema, en la cual podemos ejecutar órdenes y pasar parámetros. Si éste no es securizado existirá el riesgo de que un usuario malicioso pueda inyectar código en nuestro boot y conseguir escalar privilegios a partir de ello.

Algunas medidas de seguridad para evitar la modificación del arranque son las siguientes:

- Establecer una contraseña en la BIOS o en EFI
- Establecer una contraseña en el gestor de arranque:

Sin estas protecciones, un acceso físico ilegítimo al CPD haría que se pudiera arrancar desde una LiveCD o conseguir una shell de root, arrancando el sistema en modo monousuario.

Recientemente los fabricantes de equipos TIC han empezado a incluir en el arranque de los equipos una función llamada **Secure Boot**.

Esta función se encarga, durante el proceso de arranque, de permitir que se ejecuten sólo aquellos programas cuya firma criptográfica (hash) esté registrada previamente en el sistema. Mediante este método se evita que se ejecuten programas no verificados durante el arranque. Por tanto, las firmas criptográficas que se vayan a usar tienen que instalarse en la base de datos de firmas de cada dispositivo TIC equipado con el "Arranque Seguro" antes de que se pueda ejecutar un componente software firmado criptográficamente en esa máquina específica.

Esta característica está pensada para ser incluida en UEFI, que según parece va a ser el sustituto de la BIOS.

8.3 Sistema de archivos

Tradicionalmente, la seguridad se ha basado en el sistema de permisos de archivos y directorios para evitar la lectura o modificación de archivos por parte de usuarios que no deberían tener permiso a ellos. Rigiéndonos por el principio de menor privilegio, debería ser necesario configurar cada archivo, directorio y sistema de fichero para permitir sólo el acceso mínimo necesario para que pueda cumplir su propósito.

Sin embargo, debido a la gran cantidad de archivos que puede llegar a tener un sistema, es casi imposible asegurarse de que cada archivo en la máquina tiene sólo los permisos que necesita. Vamos a presentar una serie de restricciones en los permisos que son casi siempre apropiadas para mejorar la seguridad del sistema.

8.3.1 Particionar para aplicar distintas directivas de seguridad

Al dividir el sistema en varias particiones, cada una puede montarse con ciertas opciones y restricciones que limitan las acciones que pueden realizarse sobre los archivos de la misma.

8.3.2 Restringir el montaje y desmontaje automático de unidades extraíbles

Por defecto, la mayoría de sistemas incluyen una gran cantidad de módulos que permiten añadir y quitar sistemas de ficheros de una manera cómoda. El problema es que podría permitir a usuarios sin privilegios introducir y montar cualquier tipo de sistema de ficheros en el sistema o que un atacante pudiera comprometer un sistema a través de algún defecto en el montaje automático en la instalación de un programa. Por todo ello se recomienda tener en cuenta:

- Deshabilitar la posibilidad de arrancar desde dispositivos USB. Esto es posible modificando la configuración de la BIOS.
- Deshabilitar el montaje automático de unidades extraíbles.
- Deshabilitar la ejecución automática (autorun) en unidades extraíbles.

8.4 Gestión de copias de seguridad

De nada habría servido todo este bastionado si un fallo de disco duro hace que se pierda toda la información y el servicio quede inaccesible. Para eso es imprescindible un sistema de copias de respaldo y si el servicio que ofrece es crítico, pensar en implantar un sistema de alta disponibilidad que redunde el servicio inmediatamente en caso de que éste caiga.

9 CONCLUSIONES

Como habrá podido observar tras la lectura de este documento, todo lo expuesto se ha comentado desde un enfoque de muy alto nivel. Hemos querido adoptar este enfoque debido a que para llevar al cabo el proceso de reforzar sistemas no existe una receta maestra. Los detalles técnicos de la estrategia para implementar el bastionado dependen de cada sistema operativo, modelo o tipo de servidor. Sin embargo a grandes rasgos, si podríamos resumirlo en lo citado en este documento. Cuentas de usuarios, aplicaciones, sesiones remotas, procesos, servicios, etc. Para los administradores de sistemas esto debe ser algo básico pues es con lo que está tratando a diario. Por tanto, protegerlos y reforzar su seguridad debe ser una tarea complementaria a la de administrarlos.

Invertir tiempo y recursos en estas labores puede dar lugar a ventajas como la reducción de incidentes de seguridad, la mejora en el rendimiento del sistema debido a que disminuimos niveles inútiles de carga, permite una administración mas simple y ayuda a la identificación rápida de problemas debido a que un gran número de causas serán descartadas a priori dadas las medidas tomadas.

Quizás se trate de una labor no trivial, pero que bien llevada a la práctica, seguro que ahorra muchos problemas, ya sean provocados de forma intencionada o por error de los usuarios.

10 REFERENCIAS

- Bauer, Michael D. [2005]. Seguridad en servidores Linux. O'Reilly.
- SANS Institute InfoSec Reading Room. Hardening bastion hosts.
- SANS Institute InfoSec Reading Room. Linux kernel hardening.
- Microsoft Dynamics NAV 5.00. [2007]. Security Hardening Guide
- Ubuntu. Kernel Hardening
- Security Art Work. Malware humano. Protegiendo un servidor de los usuarios.
- ISec Lab #13. Hardening básico de Linux. Permisos y Configuraciones