
La Biblia
*del SysAdmin

Introducción

A lo largo de una serie de artículos voy a publicar los principios básicos que todo SysAdmin debe respetar, practicar y predicar a sus pares. Estos principios los he tomado (y orgullosamente puedo decir que he respetado casi en su totalidad) del artículo General SysAdmin Principles & Guidelines publicado por el Dr. Joe Chung.

Los Administradores de Sistemas, más conocidos como SysAdmins, somos a menudo los superhéroes del departamento de sistemas, aquellos destinados a "salvar el día" (muy a menudo, generalmente).

Índice

Capítulo 1: Documentación

Este primer capítulo trata el tema más importante en el trabajo de un Administrador de Sistemas la documentación.

Documentación

- Documentar absolutamente todo el trabajo realizado.
    - Mantener un registro diario de las actividades.
        - Utilizar al menos un archivo de texto plano.
        - Guardar la fecha para cada entrada.
            - Cuidar la ortografía aunque ningún colega o superior tenga acceso a la documentación. Tener una buena ortografía es una característica destacable en toda persona, especialmente en un profesional.
            - Este registro está principalmente destinado a uno mismo:
                - Es nuestra propia FAQ.
                - Es nuestra colección de tutoriales para ayudarnos y ayudar a otros a reproducir un trabajo en el futuro sobre otro sistema.
                    - Al momento de documentar una tarea es necesario preguntarse si es suficiente información para que otros puedan reproducirla.
                - Este documento puede crecer mucho al pasar el tiempo, por lo que se debe considerar rápidamente utilizar un mejor sistema de documentación, preferentemente una wiki (mejor escalable considerando que nuestro equipo puede crecer y se puede necesitar una herramienta de edición colaborativa), un blog, o ambos.
            - Este documento tiene que ser resguardado periódicamente de forma segura.
                - Algunos administradores guardan notas en papel, algo totalmente arcaico y opuesto a la tecnología con la que trabajan.
                - Si el registro/log/documentación se encuentra en un servidor corporativo dentro de una red privada y se necesita información vital estando a kilómetros de distancia, estaremos en problemas. De ser posible (siempre que no incluya información sensible/confidencial) debe ser accesible desde cualquier ubicación.
        - Utilizar comentarios en archivos de configuración para documentar los cambios.

        - Agregar una identificación del administrador (nombre) y fecha en los comentarios para saber con exactitud quién y cuándo realizó un cambio en la configuración.
- En general, debemos ser metódicos y organizados.
    - Este puede que sea uno de los aspectos más difíciles de alcanzar pues es una característica propia de la personalidad y carácter de cada persona.
    - Ya sea para organizar la jerarquía de directorios de nuestro `$HOME` o nuestra documentación, debemos mantener un orden y "limpieza digital".
    - Un SysAdmin bien organizado está mejor preparado para enfrentar situaciones adversas.
- Aprovechar el uso de la tecnología para documentar.
    - Configurar un sitio Web o FAQ (wiki, blog, etc.)
    - Hacer un uso intensivo de las imágenes y capturas de pantalla en la documentación (siempre que corresponda).
    - Utilizar medios enriquecidos (video) siempre que sea posible.
- Recordar y resguardar enlaces a todas las fuentes de información valiosa.
    - Para cada entrada, agregar todas las fuentes de información que nos ayudaron a resolver una tarea.
    - Mantener un registro centralizado (y accesible desde cualquier parte) de fuentes de información valiosa (sean tutoriales, manuales, FAQs, guías, wikis, listas de correo, foros, etc.) organizado por tópicos y tecnologías.

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 2: Lo difícil (seguridad y backups)

Este segundo capítulo trata el tema más difícil en el trabajo de un SysAdmin: seguridad y backups.

Lo difícil (seguridad y backups)

- Ocuparse primero de lo más difícil.
    - Los mayores temores de todo SysAdmin son:
        - Que un sistema sea comprometido o su seguridad vulnerada ("hackeado").
        - Que información importante se pierda o borre y no exista una copia de seguridad (backup). Los esquemas de backup, seguridad (hardening) y PRD (Plan de Recuperación ante Desastres) deben ser las primeras ocupaciones al momento de instalar/configurar un sistema o aplicación, y deben tener la mayor prioridad.
    - En ciertas empresas es necesario luchar a capa y espada (argumentos y justificaciones), todos los días, contra el mal *management* que sólo quiere que las cosas funcionen rápido y fácil, para luego "cuando haya tiempo" dedicarle un poco a la seguridad y backups.
    - Preguntarse a menudo ¿cuán bueno es nuestro esquema de backup?
        - ¿Existe hardware *spare* para reemplazar cada uno de los componentes de la infraestructura (dispositivos de energía, red, almacenamiento y servidores)?
        - En caso de utilizar tecnologías cloud, ¿se provee backup de toda la infraestructura en la nube? Si el proveedor cloud provee mecanismos de backup ¿se ha efectuado un análisis detallado del mismo? y ¿es posible auditarlo? En caso de desastres de grandes magnitudes, ¿tenemos backups disponibles para replicar toda la infraestructura cloud en otro proveedor? Claramente la tecnología cloud requiere todo un capítulo aparte, ya que abre todo un abanico de nuevas posibilidades, y la importancia de las cuestiones de backup y seguridad aumenta considerablemente.
        - ¿Existe un plan y metodología para restaurar un sistema completo desde cero (incluyendo hardware y sistema operativo)?
        - ¿Se han hecho pruebas suficientes para verificar que la información almacenada en las copias de seguridad y el mecanismo de restauración funcionan de manera efectiva? En este punto es de vital importancia que el equipo de SysAdmins tenga lugar en su agenda para al menos un simulacro anual de recuperación ante desastres (PRD). Esta tarea debe ser la prioridad #1 en el plan de trabajo anual del equipo para mantener a los procedimientos actualizados y probados.
        - Recuperar la infraestructura el día del juicio final debe ser una tarea de rutina.
    - Monitorear la correctitud e integridad de los sistemas de archivos periódicamente.
        - Los sistemas operativos suelen verificar la correctitud (fsck) de los sistemas de archivos automáticamente durante el inicio, aunque también es de vital

importancia verificar periódicamente el estado de "salud" de los discos físicos que provee la tecnología S.M.A.R.T.

- Utilizar una herramienta de comprobación periódica de integridad de los archivos, por ejemplo AIDE.
    - ¿Se protege adecuadamente la seguridad de la base de datos de la herramienta de monitoreo de integridad? ¿Se encuentra ésta almacenada en un medio de sólo lectura o accesible a través de un mount NFS de sólo lectura?
    - Si la base requiere acceso de escritura, ¿posee una configuración de permisos adecuada?
- Correr un escáner de malware y rootkits periódicamente.
    - Preferentemente desde un medio de sólo lectura.

      Respecto a este punto es posible definir un export NFS, en un servidor de seguridad centralizado, que provea acceso a todas las herramientas de seguridad necesarias en modo de sólo lectura. De esta forma se evita que las mismas puedan ser corrompidas por un atacante (para pasar desapercibido) durante un break-in.

      Esto tiene un beneficio adicional que consiste en evitar tener que replicar una herramienta de seguridad a lo largo de todos los servidores de la infraestructura, lo que simplifica la tarea de mantenerla actualizada. Sólo basta definir un subdirectorio para cada sistema operativo y arquitectura de CPU.

- Estar al tanto de las últimas vulnerabilidades, incidentes de seguridad y *exploits* descubiertos.

- Suscribirse a sitios de noticias y listas de correo que provean información de seguridad:
    - Suscribirse a las listas de correo de anuncios de seguridad de cada una de las distribuciones y sistemas operativos utilizados.
    - Suscribirse a listas de correo o feeds RSS de anuncios de seguridad de cada una de las aplicaciones utilizadas (aplicaciones Web, CMS, software de oficina, etc.)
    - Recursos:
        - www.exploit-db.com: Offensive Security Exploit Database Archive.
        - www.securityfocus.com: SecurityFocus.
        - www.helpnetsecurity.com: Help Net Security.
        - www.linuxsecurity.com: LinuxSecurity.
        - www.symantec.com: Vulnerabilities - Symantec Corp.
- Utilizar logwatch u otro software de monitoreo y análisis de archivos de log para detectar irregularidades en un sistema.
    - Ejecutar algún sistema de detección de intrusos.
    - Utilizar iptables para filtrar el tráfico entrante.
    - Implementar listas blancas para las conexiones entrantes.

    - Utilizar listas blancas para usuarios que tienen permitido el acceso por SSH, y restringir el acceso a una shell cuando no sea necesario. Autenticar con clave pública siempre que sea posible.
- Aplicar el principio de mínimo privilegio o mínimo conocimiento a rajatabla:
    - Otorgar a los usuarios los permisos mínimos y necesarios para realizar su trabajo.
    - Otorgar a los procesos y servicios los permisos mínimos y necesarios para ejecutar sus tareas.
    - Permitir el acceso desde el exterior a la menor cantidad de puertos y protocolos posibles.
    - Ofuscar banners de servicios, acceso a aplicaciones, etc.
    - Aplicar la misma metodología en cada una de las capas de un sistema, desde red y sistema operativo, hasta aplicación.
    - Tener mucho cuidado con la mentalidad "no va a pasar aquí" o "los usuarios no son tan inteligentes" que puede involucrar grandes problemas de seguridad.
- Pensar en términos de redundancia.
    - ¿Hay un backup del backup?

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines
- Cómo instalar y configurar Bacula en Debian

Capítulo 3: La eficiencia importa

Este tercer capítulo trata acerca de la eficiencia, una de las calidades del software a la que se le debe rendir tributo.

La eficiencia importa

- Las tecnologías de proxies, cachés o mirrors pueden reducir el uso de ancho de banda en instalaciones a través de Internet y en general.
- Evitar instalaciones manuales a gran escala.
    - Utilizar el clonado de sistemas siempre que sea posible.
    - Crear y mantener plantillas de sistemas y aplicaciones.
- Optimizar el uso de recursos: memoria, CPU, discos, procesos, etc.
    - Esto es de vital importancia cuando se utilizan tecnologías cloud en modo de pago por uso.
    - Aunque se cuente con exceso de recursos, se debe optimizar el uso de los mismos para contemplar todos los aspectos y escenarios posibles: picos de carga, tiempos de respuesta, espacio para backups, crecimiento y escalabilidad de los datos a lo largo del tiempo.
- Automatizar absolutamente todas las tareas que sean rutinarias o repetitivas (este punto abarca todo un capítulo en sí mismo).
- Pensar en grande (tener la escalabilidad en mente en todo momento).
    - La eficiencia es la madre de la escalabilidad: ningún sistema ineficiente puede escalar bien.
- Optimizar el uso del tiempo.
    - Saber distinguir cuándo una tarea tiene el potencial de ser rutinaria o repetitiva para invertir tiempo en su automatización.
        - Fallar en este aspecto puede provocar que se pierda tiempo automatizando una tarea que no es rutinaria ni repetitiva.
- Documentar es ganar tiempo.
    - Dejar de lado el primer mandamiento es garantía de desperdiciar considerables cantidades de tiempo a corto, mediano y largo plazo.
- Planificar el uso del tiempo.
    - Crear TODO lists.
    - Asignar prioridades (y recalcular siempre que sea necesario).
    - Aprender a postergar solicitudes que tengan baja o nula prioridad (un ejemplo es el típico escenario en el cual un usuario pide ayuda por un ratón defectuoso, en el preciso momento en el que estamos migrando a una nueva versión de una aplicación crítica en producción).
    - Combatir la procrastinación (consultar a un experto en la materia).

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 4: Administración y acceso remoto

En este cuarto capítulo se trata la administración y acceso remoto a sistemas, aspecto indispensable en el día a día de un SysAdmin.

Administración y acceso remoto

A menos que cada sistema que administremos permanezca online sólo de 8:00 a 17:00 hs. (algo muy poco probable), la administración de sistemas no es un típico trabajo de 8 a 5. La posibilidad de trabajar de manera remota es esencial y algo que todo SysAdmin debe demandar.

- Conocer los protocolos de acceso remoto más importantes, tanto gráficos como de línea de comandos.
    - SSH (*Secure Shell*)
    - VNC (*Virtual Network Computing*)
    - RDP (*Remote Desktop Protocol*)
    - X11 (*X Window System*)
    - SPICE (*Simple Protocol for Independent Computing Environments*)
- Especialmente para el caso de los administradores Unix/Linux, dominar OpenSSH y todas sus funcionalidades y características:
    - Autenticar con clave pública (sin utilizar contraseñas).
    - Transferir archivos a través de SSH puro.
    - Conocer la herramienta scp.
    - Configurar accesos SSH restringidos al protocolo SFTP.
- Montar VPNs, o utilizar IPSec, para acceder a redes corporativas de forma segura.
- Habilitar Wake-on-LAN y *Restore on AC/Power Loss* en la configuración de energía de la BIOS (ACPI) de todos los sistemas, para poder encender cualquiera de ellos sin necesidad de ingresar a la sala de servidores o centro de cómputo.
    - En caso de tener acceso o incidencia en los procesos de compra, asegurarse de que los sistemas a adquirir incluyan estas características.
- Aprender a utilizar consolas serie y switches KVM.
- Habilitar más de una forma de acceso a un sistema de forma remota.
    - En caso de bajar accidentalmente una interfaz de red (`ifdown eth0`) o cometer un error en la configuración de un firewall, quedaremos sin acceso remoto a un sistema. ¿Existe alguna otra forma de acceder a dicho sistema de manera remota cuando eso ocurra?
    - Los fabricantes de hardware más importantes han desarrollado consolas remotas de acceso directo al bus de sistema (DELL/iDRAC, HP/iLO, IBM/IMM) desde una interfaz de red especial (como si estuviésemos sentados físicamente en la consola terminal serie del equipo). Esto es de vital importancia al momento de realizar tareas críticas como migraciones en producción.
    - Para los sistemas en la nube (IaaS, ¿ofrece el proveedor más de un acceso a dichos sistemas (consolas Web, VPNs, etc.)? Nuevamente, en caso de tener incidencia en los

procesos de compras, asegurarse que estos accesos estén incluídos dentro del soporte de la infraestructura *cloud*.

- Aprender a utilizar la línea de comandos.
  - Incluso en Windows (`cmd`/PowerShell).
  - La administración remota es mejor (prácticamente exclusiva en sistemas operativos de la familia Unix) desde línea de comandos.
  - Familiarizarse con las herramientas de *networking* básicas: `telnet`, `netcat`, `nmap`, `tcpdump`, `ssh`, `scp`, `sftp`, `dig`, `ping`, `traceroute`, etc.
  - Aprender a interactuar con los protocolos de red más utilizados: HTTP, FTP, SMTP, POP3, etc.
- Aprender a implementar *port forwarding* y túneles SSH.
  - Comprender el concepto de *port forwarding* para implementar conexiones seguras entre sistemas remotos y redirigir tráfico a través de SSH.
  - Conocer el uso de túneles SSH para encriptar tráfico de protocolos planos (por ejemplo VNC).

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 5: Reinicios y uptime

Quinto episodio, trata sobre el uptime, principal meta u objetivo de un SysAdmin.

Imagen cortesía de xkcd (CC BY-NC 2.5).

Reinicios y uptime

Reinicios, también conocidos como *power cycles*, son algo trivial en un sistema personal o de escritorio. Sin embargo cuando se trata de cualquier sistema que presta servicios a terceros, ya sean usuarios u otros servicios o sistemas (por ejemplo, un servidor de bases de datos que da soporte a un sitio Web y sólo es accedido por el servidor en cuestión), un reinicio se torna en algo muy delicado.

¿Qué es el bendito "uptime"? El uptime es el tiempo que un sistema estuvo corriendo de manera ininterrumpida. Generalmente se cuenta como la cantidad de tiempo transcurrido desde el último inicio, pero también es común verlo en forma porcentual (especialmente en servicios de *hosting*, soluciones cloud, enlaces, etc.) como el porcentaje de tiempo que el sistema estuvo disponible en el transcurso de un mes. Incluso existen servicios gratuitos para monitorear el uptime de diferentes tipos de servicios.

Es necesario realizar ciertos chequeos antes de apagar o reiniciar un sistema.

- ¿Hay algún usuario logueado en el sistema?
    - En caso de servicios HTTP, ¿hay actividad en el servidor Web?. Determinar si hay usuarios y sesiones activas, y conexiones establecidas.
    - Si se trata de sistemas que proveen servicios a otros servidores, ¿hay conexiones establecidas?
- Si se debe realizar una tarea de mantenimiento crítica, como actualizaciones, migraciones o conversiones ¿durante cuánto tiempo es posible mantener el sistema apagado o el servicio inaccesible?
    - Es necesario planificar cuidadosamente un periodo de *downtime* para afectar lo menos posible el servicio, más precisamente a los usuarios del mismo.

    - Si no es posible encontrar un período de baja carga o utilización, será necesario coordinar con las áreas responsables, representantes o usuarios del servicio en cuestión para planificar la baja del mismo, en cual caso se deberá notificar el período de interrupción del servicio con su debida anticipación, a través canales de comunicación formales, contando con la autorización de los responsables o representantes del servicio.
    - De cualquier forma siempre se debe notificar una baja planificada. Nunca dar de baja un servicio o reiniciar un sistema operativo sin la debida notificación.
- ¿Existe un backup para el sistema o servicio involucrado? ¿Es necesario disponer de un backup? En tal caso, ¿se han ejecutado las pruebas de recuperación correspondientes?
    - Es de vital importancia verificar los backups y procedimientos de recuperación, especialmente para el caso crítico de migraciones y conversiones en producción.
    - Siempre planificar, probar y documentar este tipo de tareas críticas en entornos de desarrollo/testing para evitar cualquier tipo de inconvenientes ni sorpresas desagradables.
- Siempre preguntarse si es realmente necesario reiniciar un sistema.
    - En la actualidad, las tecnologías de virtualización hacen que sea cada vez menos frecuente la necesidad de un reinicio. Incluso es posible redimensionar un disco rígido sin necesidad de reiniciar el sistema.
    - ¿Estamos reiniciando el sistema/servicio para tratar de resolver un inconveniente?
        - Esta es una estrategia pobre al momento de encarar un problema, y es un comportamiento lamentablemente común en SysAdmins Jr. o con poca experiencia, la cual demuestra falta de conocimiento y metodologías ante colegas, superiores y usuarios por igual.
        - Nunca reiniciar un sistema para tratar que vuelva a la vida. Este comportamiento tiene dos consecuencias desastrosas, si tenemos la "suerte" de que el problema se solucione:
            - Primero, es probable que nunca sepamos qué ocurrió realmente ni qué fue lo que produjo el error. A veces se presentan circunstancias específicas como *deadlocks*, conflictos de recursos y otras excepciones que no quedan registradas en un log (o no es posible diagnosticar a través de los mismos). Al mismo tiempo, al reiniciar el servicio/sistema se pierden las condiciones que llevaron a producir el error. Con lo cual será imposible encontrar la solución definitiva para que el problema no vuelva a ocurrir.
            - Segundo, hemos perdido una buena oportunidad para aprender algo nuevo. A partir del momento en que resolvemos un problema reiniciando el servicio o sistema, nos convertimos en el robot que reinicia el servicio cuando falla, lo cual es lamentable y patético.
        - Por otro lado, si el problema no se soluciona, sólo perdimos tiempo y acumulamos frustración.

- Si vamos a reiniciar un sistema operativo, ¿ha sido el *bootloader* correctamente configurado?
- En caso de fallo al iniciar, ¿existe una forma alternativa para acceder al sistema? Ver el Capítulo 4 al respecto.
- Si estamos reiniciando un servicio, ¿hemos comprobado que su configuración sea correcta?
    - Es frecuente la necesidad de reiniciar un servicio cuando se cambia su configuración. En tales casos es indispensable verificar la correctitud de la configuración, especialmente su sintaxis, para evitar *downtimes* y dolores de cabeza. La mayoría de los servicios proveen herramientas para verificar la validez y correcta sintaxis de un archivo de configuración, por ejemplo para Apache y Nginx:

      ```
      apache2ctl configtest
      service nginx configtest
      ```

    - ¿Tenemos una copia de respaldo de la configuración anterior al alcance de la mano?

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 6: Automatización y scripting

Sexto capítulo dedicado al scripting y automatización de tareas. Un buen SysAdmin jamás debe repetir más de dos veces una misma tarea. Toda tarea que requiera ser llevada a cabo más de una vez amerita ser automatizada.

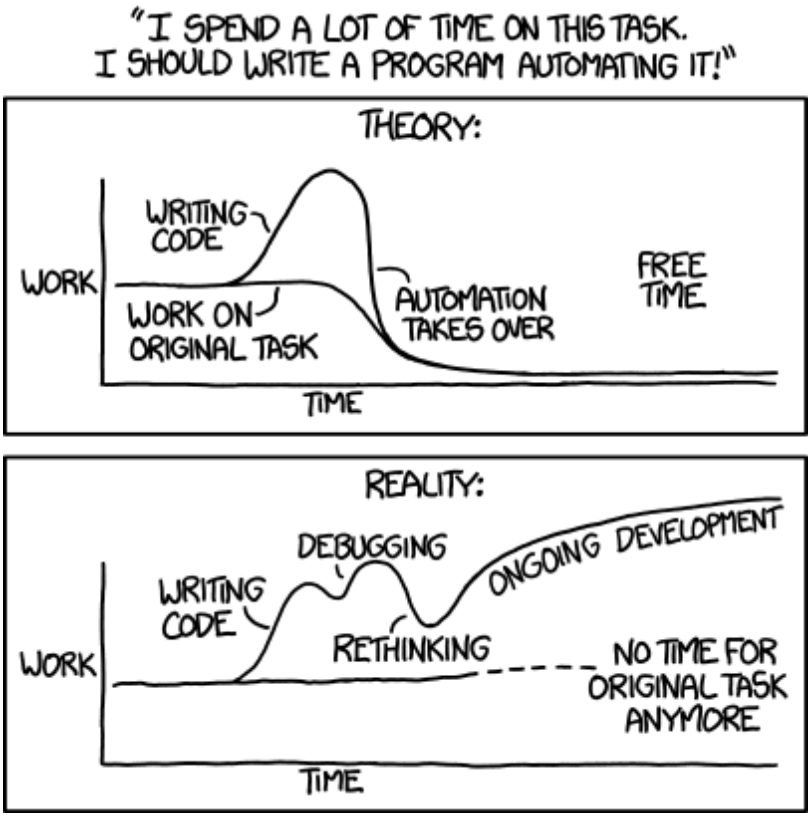

Imagen cortesía de xkcd (CC BY-NC 2.5).

Automatización y scripting

Más allá del brillante cómic de xkcd, la metodología de trabajo de un SysAdmin debería ser algo más o menos como lo siguiente. La primera vez que se ejecuta una tarea, se documenta con precisión. La segunda vez, se resuelve a partir de la documentación generada anteriormente y al mismo tiempo se automatiza de la manera más eficiente posible. La tercera vez, se delega a un robot, es decir, se deja todo en manos de la automatización.

- Aprender y utilizar de manera consistente al menos un lenguaje de scripting (Bash, Perl, Python).
    - Utilizarlo de manera consistente ayuda a perfeccionarnos y sacarle el máximo provecho.
    - Dominar más de un lenguaje nos ayuda a comprender sus diferencias, similitudes y limitaciones, lo cual sirve para saber optar por la mejor solución para cada problema.

al mismo tiempo, lenguajes interpretados como Python son más portables que Bash, lo cual nos permite atravesar fronteras en cuanto a sistemas operativos respecta.
    - Conocer y aprender la sintaxis básica de al menos un lenguaje específico de otra plataforma. Por ejemplo, si somos administradores de sistemas GNU/Linux, aprender lo básico sobre PowerShell o WSH.
    - Sin embargo no conviene especializarse en un lenguaje que sea lo suficientemente obscuro e incomprensible para que sólo uno lo entienda y utilice.
- Salvo que nos guste demasiado tipear, crear aliases y scripts cortos para reducir al máximo el uso de teclado. Incluso para las tareas tan triviales como ejecutar `apt-get update && apt-get upgrade`:

```
alias actualizar='apt-get update && apt-get upgrade'
```

- Familiarizarse con el uso de la tecla `TAB` para auto-completar y el historial (`history`, `!`) de Bash/csh.
    - Aumentar el tamaño del historial (variables de entorno `HISTFILE` y `HISTFILESIZE`.
- Procesamiento automático/periódico/programado (cron, `at`).
    - Escribir scripts para llevar a cabo tareas programadas o periódicas (`crontab`, `/etc/cron.*/`).
    - Entender el funcionamiento del entorno (`env`) y sus variables, al igual que del proceso de inicialización de una shell (Bash/csh).
    - Aprender a redirigir salidas estándar y de errores a archivos, y desarrollar scripts que lleven a cabo tareas desatendidas (no interactivas).
    - Conocer herramientas como `expect` para interactuar con procesos interactivos de manera desatendida.
    - Aprender a generar parámetros de manera dinámica con `xargs` y redirigir la entrada estándar (`<`).
    - Testear y verificar el funcionamiento de los scripts ejecutados por `cron` como cualquier otro programa. Hacer uso del `syslog` para diagnosticar errores.
- Escribir scripts que sean escalables.
    - Esto permite que un script siga siendo útil a pesar de que se sigan agregando más y más sistemas a nuestra granja.
    - Aplica a cualquier solución de administración: pensar siempre en términos de escalabilidad. Una solución se vuelve cuanto más util a medida que aplica a más y más sistemas.
    - Dominar las técnicas de paralelismo, concurrencia y subprocesos al momento de crear scripts. Esto implica conocer los comandos embebidos en las shells, tales como `jobs`, `bg`, `fg`, `&`, etc.

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 7: Gestión de software y mantenimiento del sistema

Este capítulo trata sobre una de las tareas más frecuentes en el día a día de un SysAdmin: instalar, configurar y mantener software.

Gestión de software y mantenimiento del sistema

- Instalación de software
    - Siempre utilizar fuentes oficiales para descargar software a instalar (por ejemplo: Debian, Microsoft)
        - En el caso de distribuciones GNU/Linux como Debian, la mayor parte del software puede ser instalado desde el gestor de paquetes ejecutando `apt-get` o `aptitude` sin necesidad de agregar repositorios o *mirrors* de terceros.
        - En el caso de otros sistemas operativos como Microsoft Windows, siempre descargar actualizaciones y software desde el sitio oficial.
    - Si los repositorios oficiales no están disponibles o no incluyen el software que necesitamos, utilizar una fuente o *mirror* de terceros recomendada sólo si es de confianza. Por ejemplo un repositorio mantenido por el propio proveedor del software necesario.
        - Siempre descargar e instalar (`gpg --import`) las clave de cifrado de los repositorios para asegurarse de que el software que se obtiene está firmado y es de confianza. Al mismo tiempo verificar las firmas cada vez que se descargan imágenes y software vía HTTP/S (`gpg --verify`).
            - De esta forma, los paquetes instalados deben ser mantenidos y actualizados utilizando exclusivamente el gestor de paquetes (`apt-get`, `aptitude`, `yum`, etc.)
    - Si no se dispone del software necesario en los repositorios oficiales, y no hay un repositorio externo recomendado, lo recomendable es descargar el código fuente para compilarlo e instalarlo manualmente.
        - Verificar los *checksums* cada vez que se descarga un fuente vía HTTP/S. Preferentemente descargar fuentes sólo a través de HTTPS.
        - Siempre utilizar un script para configurar, compilar e instalar un paquete. Esto permite "recordar" las opciones de configuración utilizadas para compilar el paquete, las cuales serán necesarias al momento de re-compilar el software (por ejemplo cuando se publican parches de seguridad para la versión instalada).
            - Si un software no provisto por los repositorios oficiales será utilizado en varios servidores, tal vez lo más conveniente sea crear nuestro propio paquete. Todas las distribuciones GNU/Linux poseen guías completas para generar paquetes (por ejemplo Debian Wiki).

        - Si la escala aumenta y la cantidad de paquetes personalizados es grande, será necesario montar nuestro propio repositorio (por ejemplo DebianRepository/Setup - Debian Wiki).
- Múltiples esquemas de gestión de paquetes en un mismo sistema.
    - Un ejemplo es el caso de la gestión de ports y paquetes en FreeBSD. Es recomendable mantener una cierta consistencia al momento de instalar paquetes y actualizar el sistema.
    - Generalmente los ports suelen estar más actualizados que los paquetes (a causa del tiempo necesario para compilar para todas las arquitecturas soportadas).
    - Es recomendable utilizar paquetes ya que el tiempo de compilación suele ser elevado y la ganancia en *performance* no es significativa. Por otro lado, tener una mezcla entre ports y paquetes dificulta el mantenimiento del sistema.
    - Sólo utilizar ports para disponer de opciones personalizadas en paquetes específicos.
    - Existe software que sólo es posible instalar a través de ports debido a su licencia.
- Los paquetes disponibles en los repositorios oficiales no siempre están actualizados.
    - Generalmente están actualizados en cuanto a nivel de parches de seguridad (si la versión específica aún tiene soporte).
        - Toda distribución que se precie de ser robusta publicará los parches de seguridad de paquetes, para las versiones cuyo soporte está vigente, con la mayor celeridad posible.
        - Más allá de esto, y tal como menciona el Capítulo 2, es altamente recomendable suscribirse a las listas de correo de seguridad, principalmente para planificar con anticipación las actualizaciones críticas y urgentes. Especialmente cuando se trata de actualizaciones que requieren de reinicios de servicios o, peor aún, del sistema operativo.
    - Pero las versiones oficiales suelen no estar actualizadas respecto a características y funcionalidades. Recordar que los repositorios son mantenidos por gente que tiene vida propia, muchas veces como voluntarios, haciendo este trabajo en su propio tiempo libre.
    - Cuando se requiere contar con versiones actualizadas de un paquete (sin migrar a una versión más actualizada del sistema operativo, en caso de que haya una disponible) no queda otra alternativa que compilar, instalar y mantener el paquete en cuestión manualmente.
- Actualizaciones automáticas
    - Por más lindo que suene, un Administrador de Sistemas siempre debe saber qué se está instalando o actualizando.
    - La instalación y actualización de paquetes debe ser una tarea interactiva. Muchas veces una actualización de paquetes dispara el reinicio de servicios, lo cual puede afectar el uptime de un servicio.

        - Peor aún, uno debe conocer con exactitud hacia qué versión de cada paquete se está migrando con exactitud para no romper la compatibilidad con todo software de terceros y aplicaciones a medida corriendo en el servidor.
        - Esto aplica principalmente (pero no únicamente) a las versiones de sistemas de gestión de bases de datos, lenguajes de scripting (PHP, Perl, Python), intérpretes y entornos de tiempo de ejecución (Java), servicios (Apache, Nginx), librerías del sistema (libc) y utilitarios (OpenSSL/LibreSSL).
        - Cabe destacar que esto aplica generalmente para las migraciones de versión de sistema operativo (`dist-upgrade`) más que para las actualizaciones de paquetes (`upgrade`). Al menos en Debian y derivados, donde al momento de liberar un versión estable se "congelan" las versiones de paquetes, y las subsecuentes actualizaciones sólo corrigen bugs y proveen parches de seguridad sin alterar la funcionalidad. Pero por supuesto, depende de cómo gestione el versionado de paquetes cada distribución en particular (por esta razón no se ven distribuciones *rolling-release* corriendo en servidores).
- Actualizar el sistema operativo, kernel y todo software cuidadosamente.
    - (Aplicar parches de seguridad, service packs, hotfixes, etc.)
    - Pensar siempre si es el momento correcto para hacerlo.
        - El sistema, servicio o aplicación corriendo en el servidor ¿está siendo utilizada? ¿Hay usuarios logueados?
        - La actualización ¿puede esperar o es un parche de seguridad crítico?
    - Antes de comenzar, ¿se ha realizado un backup o snapshot del sistema previamente?
    - ¿Es posible recuperar el sistema rápidamente en caso de que la actualización falle?
    - ¿Se ha testeado la actualización en un entorno de prueba antes?
    - ¿Se ha testeado el funcionamiento del sistema luego de actualizar?
    - En muchos sistemas es posible actualizar sólo los parches de seguridad o hotfixes críticos en lugar de todas las actualizaciones disponibles.
    - Y como regla general: jamás actualizar un sistema un viernes. Preferentemente no hacer nada un viernes.

- Instalar sólo software que será realmente utilizado.
    - Todo software innecesario puede contener vulnerabilidades. Sin contar con el hecho de que, al no ser utilizado, se le deje de prestar atención y quede desactualizado/obsoleto.
    - Cada pieza de software es un punto de entrada más al sistema.
    - Sin embargo, privar a los usuarios del software que necesitan puede motivarlos a instalarlo por sus propios medios, muy probablemente de manera incorrecta y en ubicaciones poco convenientes, sin mantenimiento ni monitoreo.
- No todo el software viene empaquetado de manera elegante.
    - En general sucede con el software comercial, propietario o "no gratuito".
        - Casos típicos como los parches de Microsoft, soluciones de backup de Symantec, Java de Oracle, las VMware Tools de VMware, y lamentablemente muchos más.
        - Todos los recaudos respecto al testeo, documentación y verificación de instalaciones y actualizaciones se deben potenciar para estos casos. Es común que los instaladores sean scripts Bash basados en parseo de salidas, los cuales pueden fallar horriblemente y dejar el sistema en un estado inconsistente.
        - Muchas veces incluso es necesario aplicar parches de terceros a los fuentes oficiales a fin de que funcionen en nuestras plataformas. Todo lo dicho anteriormente en cuanto a verificar la procedencia del software aplica con mayor énfasis para el caso de parches de código en software propietario.
    - En algunos casos existe software propietario de código abierto que es necesario instalar por fuera de los gestores de paquetes e incluso compilar por nuestros propios medios.
        - Especificar siempre `/usr/local` u `/opt` como directorios de instalación.

- Antes de instalarlo, pensar en la necesidad de desinstalarlo en un futuro. ¿Incluye un mecanismo o script de desinstalación?
- Documentar todas las rutas y archivos creados durante la instalación.

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 8: Usuarios y soporte

Este capítulo trata sobre tal vez el problema más complejo que deba enfrentar un SysAdmin: lidiar con usuarios y resolver sus problemas.

Usuarios y soporte

- Los usuarios difícilmente describan sus problemas con el nivel de detalle necesario para poder resolverlo o que nos permita ayudarlos. Frecuentemente se deben mantener conversaciones del siguiente estilo:
    - "Problemas con el sistema":

```
Usuario: Tengo problemas con el sistema.
  Admin: ¿Con cuál sistema?
Usuario: EL SISTEMA!!!

  Admin: ¿Qué problema tiene?
Usuario: No funciona.
```

    - "Me saltó un cartel":

```
Usuario: Me saltó un cartel
  Admin: ¿En qué aplicación? ¿Qué decía el cartel?
Usuario: En el sistema.
Usuario: No sé, está en inglés, tenía unos números.
Usuario: Ya lo cerré, no sé qué decía.
```

    - "Hola, ¿hablo con el sistema?":

```
Usuario: Hola, ¿hablo con el sistema?
  Admin: Err, no, usted está hablando con un ser humano.
```

    - "Anda lento":

```
Usuario: Anda lento
  Admin: ¿Qué es lo que funciona lento? ¿La aplicación, la red, su
computadora?
Usuario: TODO.
```

    - Pero el más simpático es el caso típico en el que un usuario lee literalmente una *bluescreen of death*:

```
Usuario: Tengo un problema con el sistema.
  Admin: Dígame, ¿qué le sucede?
Usuario: "A fatal exception 0E has occurred at 0028:C562F1B7 in
VXD..."
  Admin: Ok, está bien, deténgase, por favor.
Usuario: " ctpci9x(05) + 00001853. The current application will
be..."
```

- En el sentido inverso, se debe evitar dar detalles técnicos innecesarios o incomprensibles para el usuario al momento de resolver un problema.
    - No "aburrir" al usuario con detalles de bajo nivel que no comprende.
    - Muchas veces no es conveniente que el usuario conozca detalles de la implementación de un sistema o aplicación.
        - No sólo desde el punto de vista de la seguridad...
- A veces la tarea de un SysAdmin es mucho más dura cuando tiene que lidiar con usuarios impacientes y lamentablemente a veces irrespetuosos.
    - Sin embargo la vocación de servicio de un Admin debe ser preponderante.
    - De lo contrario deberá considerar una carrera en la rama del desarrollo (u otra similar) donde sólo deba lidiar con sistemas informáticos y a los sumo pares. Por esta razón los desarrolladores son más felices (?)
- Un SysAdmin debe contar con grandes dotaciones de paciencia y jamás de los jamases deberá faltar el respeto a ningún usuario (esto más bien debería ser una cualidad genérica de todo ser humano supongo).
    - Para el caso de los *DevOps* la tarea suele ser mucho más sencilla, pues en general interactúan con desarrolladores más que con usuarios finales.
    - Es mucho más fácil relacionarse con profesionales o usuarios con elevados conocimientos técnicos, como lo son los desarrolladores.
    - Resulta difícil "bajar a tierra" conceptos técnicos abstractos, y viceversa, al momento de comunicarse con usuarios finales.
        - Sin embargo esta habilidad se puede entrenar y practicar en seminarios o charlas, y por qué no también dictando cursos.
- Mesa de ayuda
    - Todos estos problemas se observan más frecuentemente en lo que yo llamo "la trinchera": el equipo de mesa de ayuda (*helpdesk*).
    - Si el SysAdmin es responsable por el equipo de *helpdesk* es necesario contar con un buen balance entre profesionales más duros (técnicamente hablando) y profesionales (o incluso mejor no profesionales) con un perfil más social.
    - De esta forma se cuenta con la "herramienta" adecuada para el problema indicado:

        - Enviar al perfil técnico si hay un problema con un router inalámbrico.
        - Enviar al perfil social cuando un usuario requiere ayuda con una planilla de Excel.
    - Siempre y cuando el SysAdmin pueda participar en los procesos de selección y haya recursos suficientes, claro está.
    - Capacitar constantemente al equipo de mesa de ayuda.
        - En este punto nuestra documentación es clave.
    - El equipo de mesa de ayuda debe funcionar como *proxy* de los usuarios.
        - Un usuario raramente debería contactar con un SysAdmin y mucho menos un desarrollador.
            - Si esto ocurre es porque el equipo de mesa de ayuda no está lo suficientemente capacitado, las tareas no están lo suficientemente bien documentadas, o simplemente hay un problema de organización.
- Resolver problemas
    - Se debe evitar a toda costa solicitar su contraseña al usuario para ingresar al sistema en cuestión y diagnosticar un problema. Esto no sólo está prohibido, sino que es un pecado en esta biblia.
        - Si por alguna extraña razón no queda otra alternativa, el usuario se verá forzado a cambiar su contraseña.
        - Sin embargo, una vez más, esto debe ocurrir sólo en casos de vida o muerte.
        - No es conveniente conocer una contraseña de un usuario.
            - Muchas veces los usuarios utilizan la misa contraseña en todos los sistemas, como por ejemplo su *home banking*. Por ende blanquear su contraseña no es de gran ayuda.
            - No queremos "quedar pegados" con futuros incidentes de seguridad.
    - En los sistemas de la familia Unix, siempre se puede recurrir a herramientas como `su`, `sudo` o `doas` para cambiar de usuario o ejecutar programas a nombre de otro.
    - A su vez se debe contar con usuarios de prueba en todas las aplicaciones Web bajo nuestra órbita.
    - Siempre se puede pedir una *screenshot* o al menos una captura con el móvil.
        - Este suele ser el recurso más adecuado para resolver un problema, en conjunto con la información recogida desde los logs.
        - Y hablando de logs... Los logs del sistema y aplicaciones deben ser siempre el primer lugar donde acudir en caso de problemas no reportados por usuarios.
            - Especialmente en el caso de los sistemas operativos de la familia Unix, un SysAdmin debe conocer las ubicaciones de todos los archivos de log de todas las aplicaciones y el sistema.
            - Evitar logs en formatos que no sean de texto plano y aberraciones como `journald`.

        - En otros sistemas operativos (notablemente Windows) no hay escapatoria a los logs en formato binario (se debe recurrir a herramientas nefastas como Event Viewer).
- *"Eat your own dogfood"*
    - El término "eat your own dogfood", o simplemente *dogfooding*, no tiene una traducción directa o refrán compatible en el idioma español, y se utiliza en este caso para indicar que debemos utilizar lo que les damos a los usuarios. Es decir, no debemos utilizar Windows 10 en nuestras estaciones de trabajo si a los usuarios les damos Windows 8.1, por poner un ejemplo.
    - Utilizar el mismo entorno, mismas herramientas y mismas máquinas que les damos a nuestros usuarios, en las mismas redes. Esto tiene varias ventajas:
        - Podremos identificar y resolver muchos problemas antes de que ocurran (al menos antes de que les ocurra a los usuarios).
        - Estar familiarizados con las aplicaciones y entorno de los usuarios nos permite ayudarlos de manera más eficaz y eficiente.
        - Estar familiarizados con las aplicaciones y entorno de los usuarios nos permite identificar más rápidamente la causa o raíz de un problema.
        - Al utilizar las mismas herramientas podremos conocer con exactitud sus ventajas y limitaciones.
    - Ser un usuario. Utilizar herramientas de usuario.
        - No se puede llegar a ser un administrador de sistemas Unix sin ser antes un usuario de sistemas Unix.
        - Este principio aplica a todo sistema informático.
        - Utilizar una distribución GNU/Linux en nuestra estación de trabajo y darles Windows a los usuarios sólo va a dificultar nuestra tarea de soporte.
            - Como mínimo se debe contar con una máquina virtual corriendo el mismo entorno que nuestros usuarios.
    - Preocuparse por la usabilidad del entorno de sistema operativo de los usuarios.
        - Esto tiene el beneficio extra de reducir el soporte y dolores de cabeza.
        - Utilizar una configuración sensible para los perfiles de las shells.
        - Recurrir a `sudo` para permitir cierto tipo de operaciones privilegiadas.
            - Sin comprometer la seguridad de los sistemas.
        - Utilizar una interfaz de usuario por defecto (gestor de ventanas o escritorio) familiar e intuitivo.
            - Encontrar un balance correcto entre facilidad de uso para el usuario, eficiencia y simplicidad de administración.
            - No siempre es fácil encontrar un balance entre las dos primeras,
        - Proveer a los usuarios con las herramientas necesarias para llevar a cabo sus tareas. Contar con herramientas ofimáticas adecuadas.
        - Estar abierto a las sugerencias de los usuarios en cuanto a aplicaciones, restricciones y limitaciones, manteniendo siempre el foco en la seguridad, escalabilidad y mantenibilidad de toda solución propuesta.

        - De nada sirve una aplicación que ningún usuario utilizada, por más buena que sea.
        - No es de gran ayuda proveer una solución de almacenamiento en la nube si la cuota de espacio ofrecida no es suficiente para la mayoría de los usuarios.
        - Siempre es preferible proveer una aplicación instalada y configurada correctamente a que los usuarios "se las rebusquen por sus propios medios".
            - Evitar que los usuarios recurran a software "portable".
            - Evitar que los usuarios compartan archivos a través de, o utilicen como sistema de backup a, medios removibles.
            - Sólo por citar un par de ejemplos...
- Educar y entrenar a los usuarios
    - ¿Existe documentación de alto nivel o una página Web de soporte para los usuarios?
    - Planificar capacitaciones de uso general y de aplicaciones específicas para incentivar a los usuarios.
    - Siempre es un buen momento para promover el uso de la herramienta adecuada para cada tarea.
    - Es una buena idea planificar seminarios y charlas, especialmente para concientizar acerca de la seguridad de la información y buen uso de las herramientas provistas.
- Nunca subestimar a los usuarios
    - No asumir que los usuarios no están lo suficientemente formados/capacitados o que desconocen o no son conscientes de las últimas brechas de seguridad encontradas.
    - El atacante más peligroso y más frecuente es el atacante interno.
    - Jamás caer en la mentalidad del "nunca va a pasar aquí".

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 9: Estandarización vs. diversidad

Este capítulo trata sobre estandarización y diversidad. Ser un generalista o un experto. Ser un experto es concentrar el conocimiento en un campo reducido. En otras palabras, saber mucho acerca de poco hasta el punto de saber absolutamente todo acerca de nada.

Estandarización vs. diversidad

- ¿Estandarizarse en una única distribución GNU/Linux o utilizar múltiples distros?
    - Si se opta por la estandarización, es imprescindible seleccionar tu distribución GNU/Linux cuidadosamente
        - Considerar la continuidad del proyecto como una de las claves.
            - Debian es la mejor apuesta cuando de continuidad se trata.
                - Al igual que muchas distribuciones basadas en Debian, como por ejemplo Ubuntu y Devuan.
        - Otras distribuciones poseen mayor continuidad que Debian (notablemente Slackware), sin embargo se debe ponderar a su vez el tamaño de la comunidad y especialmente el tamaño del equipo de desarrollo.
        - Un tercer criterio a considerar es qué empresa está detrás de, y qué intereses persigue, la dirección del proyecto.
    - Si optamos por la estandarización, puede que ocurra que no sepamos resolver una tarea tan simple como configurar la red en una distro fuera de la que utilizamos a diario.
    - Utilizar múltiples distribuciones es una buena (¿mejor?) idea.
        - Red Hat y sus derivados siguen siendo los más ampliamente utilizados en entornos corporativos.
            - Esto implica que es necesario estar familiarizado con Red Hat y el gestor y formato de paquetes RPM.
        - Otras distribuciones muy utilizadas son aquellas basadas en SUSE.
        - Contar con una buena flexibilidad en cuanto a distribuciones y gestores de paquetes puede potenciar nuestras oportunidades al momento de considerar un cambio de rumbo o nuevas oportunidades laborales en nuestra carrera profesional.
        - Jamás caer en el distrohopping. No es broma y podría considerarse una enfermedad. Reinstalar distros una y otra vez no nos hace expertos en Linux e involucra una gran pérdida de tiempo. El aprendizaje está en el uso y configuración de una distro, instalar y ver barras de progreso no aporta nada útil.
        - Utilizar distribuciones estables y robustas en nuestra estación de trabajo.
            - No poder realizar una tarea, conectarse a un servidor o abrir un archivo a causa de un problema con nuestra estación de trabajo (ya sea por un fallo o por falta de soporte para alguna aplicación o protocolo)

no sólo es vergonzoso para un SysAdmin, sino que hace mala publicidad a nuestra querida distro.

- A lo largo de los años he visto a colegas linuxeros pasar por situaciones como las siguientes en la oficina:
  - "No puedo abrir archivos con el formato propietario .xyz".
  - "No tengo codec para el formato X".
  - "Esperá que justo está compilando una actualización y no puedo usar la máquina".
  - "Esperá que no bootea porque la última actualización rompió el entorno de escritorio".
  - "No puedo usar la aplicación X porque se actualizó la libXYZ y dejó de funcionar".
  - "La última actualización del programa X rompió la compatibilidad con el protocolo ABC".
  - Y muchas similares...
- Lamentablemente he visto muy seguido estas situaciones, especialmente con distribuciones *rolling release*, las cuales hacen parecer a GNU/Linux como un SO inestable y poco confiable. Y todos sabemos que es justamente lo opuesto, sólo que "tu distro" apesta.

- Por supuesto, siempre vamos a querer experimentar. En tal caso dejar las distros "espartanas" y *rolling release* para casa y nuestras computadoras personales.

- Utilizar múltiples sistemas operativos es todavía una mejor idea.
  - Contar con, al menos, un conocimiento básico sobre otros sistemas operativos de la familia Unix.
    - FreeBSD y OpenBSD son excelentes alternativas a GNU/Linux al momento de montar un nuevo servidor.
      - Especialmente OpenBSD está orientada a la seguridad. Es una gran opción para montar firewalls.
      - FreeBSD cuenta de forma nativa con ZFS. La mejor opción para servidores de archivos (NFS, Samba, etc.)
    - Existen muchas otras alternativas Unix como Solais, AIX, y *BSD (NetBSD, DragonFly).
  - No se puede ser un completo ignorante respecto a los sistemas operativos Microsoft Windows.
    - En toda organización existe al menos un servidor Windows.
    - Soluciones como Active Directory suelen ser superadoras y altamente adoptadas en entornos corporativos.
- Como en todo aspecto de la vida, estandarizarse nos deja "pegados" a una tecnología en particular (es por ello la importancia en la elección). Al contrario la diversidad nos da

flexibilidad, tolerancia a imprevistos, mayor sabiduría (por tener un panorama más claro y ampio respecto a los cambios de comportamiento en el mercado o la comunidad), entre otras cualidades. Todo esto al costo de no llegar a profundizar al máximo nuestros conocimientos en cada uno de los temas que abarcamos.

- El secreto está en encontrar un balance entre ambas.
- Se supone que un SysAdmin debe conocer todo acerca de todo. Ya sabemos que es imposible, pero es lo que la sociedad espera de nosotros.

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Capítulo 10: Promoción, ética y aprendizaje

Este capítulo final trata sobre algunos aspectos no estrictamente relacionados con lo técnico, pero si lo profesional.

Promoción, ética y aprendizaje

Promoción de GNU/Linux

- ¿Debe un Administrador GNU/Linux promocionar el uso de nuestro sistema favorito?
    - Hasta un cierto punto.
        - Aunque si es importante dar a conocer la filosofía del software libre.
            - Sin ser un fundamentalista del software libre.
    - Un SysAdmin debe promocionar el uso de la herramienta adecuada para cada tarea...
        - ...y conocer y aprender la herramienta adecuada para cada tarea en particular. Incluso si resulta ser Windows.
    - No caer en la tentación de "convertir" usuarios.
        - Cada sistema de cada usuario/colega/amigo/familiar convertido a GNU/Linux pasará a ser automáticamente nuestra responsabilidad.
        - Un SysAdmin suele estar lo suficientemente ocupado como para hacerse cargo de problemas adicionales en su (poco) tiempo libre.
        - Como dice Paul McCartney: *"live and let die"*.

Aprendiendo Administración

- La mejor forma de aprender administración es haciendo administración.
- Es más probable que se aprenda más cuando algo falla.
- Escuchar y consultar (bombardear a preguntas) al viejo SysAdmin.
    - Generalmente los viejos SysAdmins son ávidos para compartir conocimiento. Sólo hace falta interrogarlos e interesarse por su trabajo.

Estar en contacto con nuestros colegas

- Estar al día de las últimas tendencias en cuanto a tecnología.
    - ¿Qué mejoras incorporan las nuevas versiones de los productos/sistemas/aplicaciones que utilizamos?
    - ¿Existen soluciones innovadores o cambios en la industria que aún no hemos adoptado?
        - Virtualización para aprovechar mejor nuestro hardware.
        - Cloud computing (IaaS/SaaS/PaaS) en lugar de nuestra propia infraestructura (*on premises*).

        - Soluciones híbridas.
    - ¿Existen nuevas alternativas de software para proveer servicios?
        - LDAP en lugar de NIS.
        - Samba en lugar de CIFS.
        - Nginx en lugar de Apache.
        - Cualquier cosa (?) en lugar de PHP.
- ¿Qué tecnologías usan en otras empresas o instituciones?
- ¿Estamos atrasados en algún aspecto? ¿Qué se puede cambiar o mejorar?
    - Ya sea para dar un mejor servicio como para simplificar la administración.
- ¿Existen nuevas metodologías de trabajo?
    - Software de backup vs. mis propios scripts.
    - Software de automatización vs. mis propios scripts.
    - Contenedores vs. máquinas virtuales.

Ética laboral y licencias

- Un SysAdmin jamás debe utilizar ni permitir el uso de software pirata.
    - Al ser quien gestiona el software, es el responsable por respetar las licencias de software.
    - Si un usuario desea utilizar cierto software propietario, se deberá adquirir la licencia en caso que corresponda.
        - De lo contrario no se podrá utilizar dicho software.
    - Más allá de adquirir una licencia se deberán respetar los términos de la misma.
        - Un usuario adquiere una licencia para un equipo pero desea utilizarla en varios, violando los términos de la misma.

        - Un usuario posee copias de software pirata (ya sea instalado o copias de los instaladores) en sistemas bajo nuestra órbita.
    - Por otro lado se debe garantizar que los datos alojados en nuestro servidores no estén sujetos a copyright.
        - Un usuario tiene cientos de archivos MP3 o películas pirata en uno de nuestros servidores de archivos (Samba, NFS) o soluciones de almacenamiento en la nube (Nextcloud).
        - Lo mismo ocurre para archivos PDF o EPUB (copias de libros pirata).
    - Desde el punto de vista de la ética profesional, siempre se deberá mantener dentro de la ley.
        - Un Jefe o Gerente desea acceder a la casilla de correo de un usuario.
            - Más allá de lo que puedan decir los términos de uso de los sistemas de la compañía, acceder a una cuenta de correo electrónico es ilegal en la mayoría de los países.
            - Lo mismo podría ocurrir para otro tipo de solicitudes similares.
            - Siempre estar al tanto de las cuestiones legales que afectan nuestro trabajo.
        - Como SysAdmin uno tiene naturalmente acceso a todas las cuentas de correo, mensajes, archivos y bases de datos de la compañía.
            - Tener acceso como root/Administrator nos permite (valga la redundancia) acceder a absolutamente todos los datos de la compañía.
            - Jamás utilizar esto a nuestro favor, en perjuicio de otros, o simplemente para curiosear.
            - Más allá de ser poco ético, en la mayoría de los casos se nos puede acusar de espionaje.
            - Un SysAdmin jamás debe cruzar esa barrera, bajo ninguna circunstancia, ya sea para beneficio propio o por una solicitud "de más arriba".
                - En tal caso siempre es mejor renunciar que tener una mala reputación y convertirse en un mal profesional y persona. O peor aún "quedar pegado" en un proceso judicial.
        - Un SysAdmin debe ser absolutamente responsable con las credenciales y roles que genera y/o le son otorgados.

De esta forma concluye esta Biblia del SysAdmin (con el perdón de la blasfemia para los creyentes) sin descartar la posibilidad de agregar anexos en un futuro (en el caso de que sea necesario). Próximamente publicaré una versión en PDF de la misma, para imprimir y guardar en la mesa de luz de todo SysAdmin que se precie :)

Y por qué no también una versión en formato EPUB para los E-Readers...

Esta Biblia del SysAdmin queda disponible públicamente en GitHub y abierta a sugerencias, cambios y correcciones. Al mismo tiempo es liberada bajo la licencia MIT.

Referencias

- Joe Chung - General SysAdmin Principles & Guidelines

Licencia