Administración de Sistemas Operativos

Unidad 1
Presentación Nº2
Sudo, visudo, etc/sudoers
Prof. Luis E. Fagúndez

Bibliografía

- **Web para complementar:**
  - https://www.sudo.ws/
  - http://go2linux.garron.me/sudoers-man-page/
  - **https://www.linuxtotal.com.mx/index.php?cont=info_admon_014**
  - **https://www.sudo.ws/man/1.8.15/sudoers.man.html**
  - **Última fecha de consulta de los materiales: 15/07/2019**

Repaso

- **$ su** : Comando que nos permite acceder al root pero usando las variables de entorno de nuestro usuario.
- **$ su nombreUsuario** : Comando que nos permite acceder a un “x” usuario con todos sus permisos y variables de entorno.
- **$ su -** : Comando que nos permite tener todos los permisos de root incluyendo sus variables de entorno.

Repaso

- **El directorio /sbin:**
  - Es para ejecutables de uso exclusivo por el superusuario, y solamente los necesarios para arrancar y montar el directorio /usr y ejecutar operaciones de restablecimiento del sistema.
  - Según Filesystem Hierarchy Standard: contiene típicamente ficheros indispensables para el arranque del sistema, además de los binarios contenidos en /bin.
  - Todo lo que se ejecute después que /usr es montado (cuando no hay problemas) debería estar ubicado en /usr/bin.
  - Los binarios de administración exclusivamente del sistema local deben colocarse en /usr/local/sbin.

Repaso

- **Como mínimo, en /sbin debieran estar los siguientes programas:**
  - clock, getty, init, update, mkswap, swapon, swapoff, halt, reboot, shutdown, fdisk, fsck.*, mkfs.*, lilo, arp, ifconfig, route.

“sudo”

- **Para explicar el concepto de super usuario lo dividiremos en 3 partes:**
  - $ sudo: comando con permisos de SUID, que los usuarios usan para ejecutar otros comandos a los que se les permite usar.
  - $ visudo: comando que permite al administrador modificar /etc/sudoers.
  - /etc/sudoers: el archivo que contiene las configuraciones que le indica a sudo los permisos de comandos para usuario.

“sudo”

- **Requiere que los usuarios se autentifiquen a si mismos a través de su contraseña para permitirles la ejecución del comando, ejemplo:**
  - $ sudo yum – y update
    - (Le permite a un usuario de Fedora/RedHat actualizar el sistema)
- **Por defecto, después de ejecutar el comando, se tendrá 5 minutos para volver a usar el mismo comando u otros a los que el usuario tenga permisos, sin necesidad de ingresar la contraseña de nuevo.**

“sudo”

- Si queremos saber que comandos podemos ejecutar con sudo podemos verificarlo con el modificador -l
  - $ sudo -l
- También es posible ejecutar comandos de otros usuarios del sistema indicando la opción -u
  - $ sudo -u clotilde /comando/de/clotilde

“sudo”

- **El comando sudo también permite editar archivos de texto de root (con el permiso previamente otorgado en 'sudoers'), y esto se logra con la opción -e, esta opción esta ligada a otro comando de sudo llamado “sudoedit” que invoca al editor por defecto del usuario, que generalmente suele ser vi.**
  - sudo -e /etc/inittab
    - (Permite modificar el archivo inittab como root)

“sudo”

- Cuando se configura sudo se tienen múltiples opciones que se pueden establecer, estás se consultan a través de la opción -L
  - $ sudo -L

“sudo”

- Con la opción -V podemos listar las opciones establecidas por defecto para sudo, todos los usuarios, comandos, equipos, etc.
- Este modificar es necesario ejecutarlo como root.
  - # sudo -V
- "Path to log file: /var/log/sudo.log": log del sistema “bitacora” definida por defecto para el comando sudo.
- En este archivo se almacena absolutamente todo lo que se haga con sudo, que usuarios lo usaron y con que, intentos de uso, etc.

“visudo”

- Permite la edición del archivo de configuración de sudo sudoers.
- Invoca al editor que se tenga por defecto que generalmente es vi.
- Visudo cuando es usado, bloquea el archivo /etc/sudoers de tal manera que nadie más lo puede utilizar, esto por razones obvias de seguridad que evitarán que dos o más usuarios administradores modifiquen accidentalmente los cambios que el otro realizó.

“visudo”

- IMPORTANTE: al cerrar el archivo USANDO el comando visudo, se verifica automáticamente que el archivo este bien configurado.
- Si hay errores de sintaxis principalmente en sus múltiples opciones o reglas de acceso que se tengan aparecerá un mensaje de error que nos mostrará la línea donde se encuentra, y la pregunta¿What now?.
- Por esta razón no debe editarse /etc/sudoers directamente (perfectamente posible ya que es un archivo de texto como cualquier otro) siempre se debe de usar visudo.

“visudo”

- **¿What now?**
- **Cuando aparece ésta opción tendremos 3 opciones para seguir trabajando:**
  - e - edita de nuevo el archivo, colocando el cursor en la línea del error (si el editor soporta esta función.)
  - x - salir sin guardar los cambios.
  - q - salir y guarda los cambios.

“visudo”

- **Si se desea comprobar que /etc/sudoers esta bien configurado se usa la opción -c,**
- **Toma el archivo de configuración por defecto si no se indica algún otro y realiza un chequeo.**
  - # visudo -c

“visudo y más”

- Por defecto el archivo de configuración es /etc/sudoers pero se pueden editar otros archivos que no sean ese y que se aplique la sintaxis de sudo.
- Esto se logra con la opción -f
  - # visudo -f /otro/archivo.

“visudo” - Modo estricto

- La opción -s activa el modo 'estricto' del uso de visudo.
- En éste modo se comprobará los errores sintácticos y el orden correcto de las reglas.
- Por ejemplo si se define el alias para un grupo de comandos y este se usa antes de su definición, con esta opción se detectará este tipo de errores.

/etc/sudoers

- Es el archivo de configuración de sudo.
- Ubicado bajo /etc.
- Se modifica a través del uso de visudo.
- En este archivo se establece quien (usuarios) puede ejecutar que (comandos) y de que modo (opciones).
- Se generando una lista de control de acceso que puede ser tan detallada como se desee.

/etc/sudoers

- Su configuración se divide en 3 posibles partes:
- Alias.
- Opciones (Defaults).
- Reglas de acceso.

Alias

- **Un alias se refiere a un usuario, un comando o a un equipo.**
- **El alias engloba bajo un solo nombre (nombre del alias) una serie de elementos que después en la parte de definición de reglas serán referidos aplicados bajos cierto criterio.**
- **La forma para crear un alias es la siguiente:**
  - tipo_alias NOMBRE_DEL_ALIAS = elemento1, elemento2, elemento3, … elementoN
  - tipo_alias NOMBRE1 = elemento1, elemento2 : NOMBRE2 = elemento1, elemento2

Alias

- **El tipo_alias define los elementos que se van a trabajar.**
- **Los tipo de alias son cuatro y son los siguientes:**
  - Cmnd_Alias - define alias de comandos.
  - User_Alias - define alias de usuarios normales.
  - Runas_Alias - define alias de usuarios administradores o con privilegios.
  - Host_Alias - define alias de hosts o equipos.

Alias

- El NOMBRE_DEL_ALIAS puede llevar letras, números o guión bajo (_) y debe de comenzar con una letra mayúscula, se acostumbra a usarlos siempre en mayúsculas.
- Los elementos del alias varían dependiendo del tipo de alias.

Alias: Cmnd_Alias

- **Definen uno o más comandos y otros alias de comandos que podrán ser utilizados después en alias de usuarios.**
- **Ejemplos**
  - Cmnd_Alias WEB = /usr/sbin/apachectl, /usr/sbin/httpd, sudoedit /etc/httpd/
- **Indica que a quien se le aplique el alias “WEB” podrá ejecutar los comandos apachectl, httpd y editar todo lo que este debajo del directorio /etc/httpd/**
- **Los directorios deben de terminar con '/'.**
- **Se debe de indicar la ruta completa de los comandos.**

Alias: Cmnd_Alias

- Cmnd_Alias APAGAR = /usr/bin/shutdown -h 23\:00

- **Al usuario que se le asigne el alias “APAGAR” podrá hacer uso del comando 'shutdown' exactamente con los parámetros como están indicados, es decir apagar -h (halt) el equipo a las 23:00 horas.**
- **Es necesario escapar el signo :, así como los símbolos ' : , = **

Alias: Cmnd_Alias

  - Cmnd_Alias NET_ADMIN = /sbin/ifconfig, /sbin/iptables, WEB

- **NET_ADMIN es un alias con los comandos de configuración de interfaces de red ifconfig y de firewall iptables.**
- **Además le agregamos un alias previamente definido que es WEB, así que a quien se le asigne este alias podrá hacer uso de los comandos del alias WEB.**

Alias: Cmnd_Alias

- Cmnd_Alias TODO_BIN = /usr/bin/, !/usr/bin/rpm

- **A quien se le asigne este alias podrá ejecutar todos los comandos que estén dentro del directorio /usr/bin/ menos el comando 'rpm' ubicado en el mismo directorio.**
- **NOTA IMPORTANTE: este tipo de alias con un permiso muy amplios menos '!' algo, generalmente no son una buena idea, ya que comandos nuevos que se añadan después a ese directorio también podrán ser ejecutados, es mejor siempre definir específicamente lo que se requiera.**

Alias: User_Alias

- **Definen a uno o más usuarios, grupos del sistema (indicados con %), grupos de red (netgroups indicados con +) u otros alias de usuarios.**
- **Ejemplos:**
  - User_Alias MYSQL_USERS = andy, marce, juan, %mysql
- **Indica que al alias MYSQL_USERS pertenecen los usuarios indicados individualmente más los usuarios que formen parte del grupo 'mysql'.**
  - User_Alias ADMIN = sergio, ana
- **'sergio' y 'ana' pertenecen al alias ADMIN.**

Alias: User_Alias

  - User_Alias TODOS = ALL, !samuel, !david

- **Aquí encontramos algo nuevo, definimos el alias de usuario TODOS que al poner como elemento la palabra reservada 'ALL' abarcaría a todos los usuarios del sistema, pero no deseamos a dos de ellos, así que negamos con '!', que serían los usuarios 'samuel' y 'david'.**
- **Es decir, todos los usuarios menos esos dos.**
- **NOTA IMPORTANTE: este tipo de alias con un permiso muy amplios menos '!' algo, generalmente no son una buena idea, ya que usuarios nuevos que se añadan después al sistema también serán considerados como ALL, es mejor siempre definir específicamente a los usuarios que se requieran.**
- **ALL es válido en todos los tipos de alias.**

Alias: User_Alias

  - User_Alias OPERADORES = ADMIN, alejandra

- **Los del alias ADMIN más el usuario 'alejandra'.**

Alias: Runas_Alias

- **Funciona exactamente igual que User_Alias, la única diferencia es que es posible usar el ID del usario UID con el caracter '#'.**
  - Runas_Alias OPERADORES = #501, fabian
- **Al alias OPERADORES pertenecen el usuario con UID 501 y el usuario 'fabian'.**

Alias: Host_Alias

- **Definen uno o más equipos u otros alias de host.**
- **Los equipos pueden indicarse por su nombre (si se encuentra en /etc/hosts) por nombre de dominio, si existe un solucionador de dominios, por dirección IP, por dirección IP con máscara de red. Ejemplos:**
  - Host_Alias LANS = 192.168.0.0/24, 192.168.0.1/255.255.255.0
- **El alias LANS define todos los equipos de las redes locales.**
  - Host_Alias WEBSERVERS = 172.16.0.21, web1 : DBSERVERS = 192.168.100.10, dataserver
- **Se define dos alias en el mismo renglón: WEBSERVERS y DBSERVERS con sus respectivas listas de elementos.**
  - El separador ':' es válido en cualquier definición de tipo de alias.

Opciones (defaults)

- **Las opciones o “defaults” permiten definir ciertas características de comportamiento para los alias previamente creados para**
  - Usuarios
  - usuarios privilegiados
  - para equipos
  - manera global para todos.
- **No es necesario definir opciones o defaults, sudo ya tiene establecidas el valor de cada uno, y es posible conocerlas a través de “sudo -V”.**

Opciones (defaults)

- **La potencia de sudo está en su alto nivel de configuración, por eso es importante conocer como establecer opciones especificas.**
- **Las opciones o defaults es posible establecerlos en cuatro niveles de uso:**
  - De manera global, afecta a todos.
  - Por usuario.
  - Por usuario privilegiado.
  - Por equipo (host).

- **Se usa la palabra reservada 'Defaults' para establecer las opciones y dependiendo del nivel que deseamos afectar su sintaxis es la siguiente:**
  - Global: Defaults opcion1, opcion2 ...
  - Usuario: Defaults:usuario opcion1, opcion2 ...
  - Usuario Privilegiado: Defaults>usuario opcion1, opcion2 ...
  - Equipo: Defaults@equipo opcion1, opcion2 ...

- **Los defaults que lista el man para el archivo sudoers son cuatro:**
  - flags o booleanos.
  - Enteros.
  - Cadenas.
  - Listas.

flags o booleanos

- Generalmente se usan de manera global, simplemente se indica la opción y se establece a 'on' para desactivarla 'off' se antepone el símbolo '!' a la opción.
- Es necesario consultar el manual para saber el valor por defecto 'on' o 'off' para saber si realmente necesitamos invocarla o no.

flags o booleanos: ejemplos

  - Defaults mail_always

- **Establece a 'on' la opción 'mail_always' que enviara un correo avisando cada vez que un usuario utiliza “sudo”.**
- **Esta opción requiere que 'mailto_user' este establecida.**

flags o booleanos: ejemplos

  - Defaults !authenticate, log_host
- **Desactiva 'off' el default 'authenticate' que por defecto esta activado 'on' e indica que todos los usuarios que usen sudo deben identificarse con su contraseña.**
  - Esto es un ejemplo y sería una pésima idea usarlo realmente, ya que ningún usuario necesitaría autenticarse, esto es porque estamos usando Defaults de manera global.
- **La segunda opción 'log_host' que por defecto está en 'off' la activamos y guarda el nombre del host cuando se usa un archivo (en vez de syslog) como bitácora de sudo.**

flags o booleanos: ejemplos

  - Defaults:ana !authenticate

- **Usamos opciones por usuario en vez de global, indicando que el usuario 'ana' no requiera autenticarse, pero todos los demás si.**

flags o booleanos: ejemplos

  - Defaults>ADMIN rootpw

- **Opciones para usuarios privilegiados, en vez de usar una lista de usuarios, usamos un alias 'ADMIN' que se supone fue previamente definido, y establecemos en 'on' la opción 'rootpw' que indica a sudo que los usuarios en el alias 'ADMIN' deberán usar la contraseña de 'root' en vez de la propia.**

Enteros

- Tal como su nombre lo indica, manejan valores de números enteros en sus opciones, que deben entonces usarse como opción = valor.

Enteros: ejemplos

  - Defaults:fernanda, regina passwd_tries = 1, passwd_timeout = 1

- **En este caso se establecen opciones para los usuarios 'fernanda' y 'regina' solamente, que solo tendrán una oportunidad de ingresar la contraseña correcta 'passwd_tries' el valor por defecto es de 3 y tendrán un minuto para ingresarla 'passwd_timeout' el valor por defecto son 5 minutos.**
- **La mayoría de las opciones de tiempo o de intentos, al establecerlas con un valor igual a cero entonces queda ilimitado la opción.**

Enteros: ejemplos

  - Defaults@webserver umask = 011

- **Se establecen opciones solo para los usuarios que se conectan al servidor 'webserver' y el valor 'umask' indica que si mediante la ejecución del comando que se invoque por sudo es necesario crear archivos o directorios, a estos se les aplicará la máscara de permisos indicada en el valor de la opción.**

Cadenas

- **Son valores de opciones que indican mensajes, rutas de archivos, etc.**
- **Si hubiera espacios en el valor es necesario encerrar el valor entre comillas dobles (" ").**
  - Defaults badpass_message = "Intenta de nuevo: "
- **Para todos los usuarios, cuando se equivoquen al ingresar la contraseña, es el mensaje que saldría. En este caso la opción por defecto es "Sorry: try again".**

Listas

- Permite establecer/eliminar variables de entorno propias de sudo.
- Los 'Defaults' para variables es de los menos usados en las configuraciones de sudo y ciertamente de los más confusos.
- Para entender como se aplican es más fácil si primero ejecutas como 'root' el comando sudo -V, y al final del listado encontrarás en mayúsculas las posibles variables de entorno que se pueden establecer o quitar y que vienen del shell.

Listas

- Solo existen tres opciones de listas:
- env_check
- env_delete
- env_keep
- Las listas pueden ser remplazadas con '=', añadidas con '+=', eliminadas con '-=' o deshabilitadas con '!'. Con un par de ejemplos quedará más claro.

Listas: ejemplos

- Defaults env_delete -= HOSTNAME

- **Elimina la variable de entorno 'HOSTNAME', (pero preserva todas las demás que hubiera)**
- **CUIDADO: comandos que se ejecuten bajo sudo y que requieran de esta variable no la tendrán disponible.**

Listas: ejemplos

- Defaults env_reset
- Defaults env_check += DISPLAY, PS1

- **La primera opción 'env_reset' reinicia las variables de entorno que sudo utilizará o tendrá disponibles.**
- **Solo quedan disponibles LOGNAME, SHELL, USER y USERNAME.**
- **La siguiente línea indica que agregue (+=) a lo anterior, también la variable de entorno DISPLAY a su valor establecido antes del reset.**

Reglas de acceso

- Aunque no es obligatorio declarar alias, ni opciones (defaults), y de hecho tampoco reglas de acceso, pues el archivo /etc/sudoers no tendría ninguna razón de ser si no se crean reglas de acceso.
- De hecho podríamos concretarnos a crear solamente reglas de acceso, sin opciones ni alias y podría funcionar todo muy bien.

Reglas de acceso

- Las reglas de acceso definen que usuarios ejecutan que comandos bajo que usuario y en que equipos.

Reglas de acceso: ejemplos

  - usuario host = comando1, comando2, ... comandoN

- **Sintaxis básica, 'usuario' puede ser un usuario, un alias de usuario o un grupo (indicado por %), 'host' puede ser ALL cualquier equipo, un solo equipo, un alias de equipo, una dirección IP o una definición de red IP/máscara, 'comandox' es cualquier comando indicado con su ruta completa.**
- **Si se termina en '/' como en /etc/http/ entonces indica todos los archivos dentro de ese directorio.**

Reglas de acceso: ejemplos

- **nombre_usuario nombre_equipo = (usuario:grupo) comando_restringir**

- **nombre_usuario**: Es el nombre de usuario que puede usar el comando sudo. El nombre de usuario puede ser un usuario, un alias de usuario o un grupo.
- **nombre_equipo**: Es el nombre del equipo o hostname en el que podemos aplicar el comando sudo. Sus valores pueden ser todos los equipos (ALL), un solo equipo, un alias de equipos, una dirección IP, etc.
- **(usuario:grupo)**: Especificamos los usuarios y los gruposs que podrá usar el usuario de sudo cuando ejecuta los comandos. Para poder ejecutar comandos con un usuario y grupo específico tendremos que usar los comandos sudo -u y sudo -g. Si no indicamos ningún usuario ni ningún grupo se usará el usuario root y el grupo root.
- **comando_restringir**: Especificación/restricción de los comandos que pueden ejecutar los usuarios que pueden ejecutar el comando sudo.

Reglas de acceso: ejemplo

  - daniela ALL = /sbin/iptables

- **Usuario 'daniela' en cualquier host o equipo puede utiliar iptables.**
  - ADMIN ALL = ALL
- **Los usuarios definidos en el alias 'ADMIN' desde cualquier host pueden ejecutar cualquier comando.**

Reglas de acceso: ejemplos

  - %gerentes dbserver = (director) /usr/facturacion, (root) /var/log/*

- **Los usuarios que pertenezcan al grupo del sistema llamado 'gerentes' pueden en el equipo llamado 'dbserver' ejecutar como si fueran el usuario 'director' la aplicación llamada 'facturación', además como usuarios 'root' pueden ver el contendido de los archivos que contenga el directorio /var/log.**

Reglas de acceso: ejemplos

- Lo anterior introduce algo nuevo, en la lista de comandos es posible indicar bajo que usuario se debe ejecutar el permiso.
- Por defecto es el usuario 'root', pero no siempre tiene que ser así
- Además la lista 'hereda' la primera definición de usuario que se indica entre paréntesis ( ), por eso si se tiene más de alguno hay que cambiar de usuario en el comando conveniente, el ejemplo anterior también sería válido de la siguiente manera:
  - %gerentes dbserver = /var/log/*, (director) /usr/facturacion
- No es necesario indicar (root) ya que es el usuario bajo el cual se ejecutan los comandos por defecto.
- También es válido usar (ALL) para indicar bajo cualquier usuario.

Reglas de acceso: ejemplos

- **El ejemplo siguiente da permisos absolutos.**
  - Clotilde ALL = (ALL) ALL
- **Se establece permiso para el usuario 'Clotilde' en cualquier host, ejecutar cualquier comando de cualquier usuario, por supuesto incluyendo los de root.**

Reglas de acceso: ejemplos

  - SUPERVISORES PRODUCCION = OPERACION

- **Una regala formada solo por alias.**
- **En el alias de usuario 'SUPERVISORES' los usuarios que estén indicados en ese alias, tendrán permiso en los equipos definidos en el alias de host 'PRODUCCION', de ejecutar los comandos definidos o listados en el alias de comandos 'OPERACION'.**
- **En este último ejemplo se aprecia lo útil que pueden ser los alias, ya que una vez definida la regla, solo debemos agregar o eliminar elementos de las listas de alias definidos previamente.**
- **Es decir, se agrega un equipo más a la red, se añade al alias 'PRODUCCION', un usuario renuncia a la empresa, alteramos el alias 'SUPERVISORES' eliminándolo de la lista, etc.**

Reglas de acceso: ejemplos

  - clotilde ALL = /usr/bin/passwd *, !/usr/bin/passwd root

- **Al usuario 'clotilde', desde cualquier equipo, tiene permiso de cambiar la contraseña de cualquier usuario (usando el comando 'passwd'), excepto '!' la contraseña del usuario 'root'.**
- **Lo anterior se logra mediante el uso de argumentos en los comandos.**
- **En el primer ejemplo '/usr/bin/passwd *' el asterisco indica una expansión de comodin (wildcard) que indica cualquier argumento, es decir, cualquier usuario.**
- **En el segundo caso '!/usr/bin/passwd root', si indica un argumento específico 'root', y la '!' como ya se sabe indica negación, negando entonces el permiso a cambiar la contraseña de root.**

Reglas de acceso: ejemplos

- **Cuando se indica el comando sin argumentos: /sbin/iptables sudo lo interpreta como 'puede usar iptables con cualquiera de sus argumentos'.**
  - mariajose ALL = "/sbin/lsmod"
- **Al estar entre comillas dobles un comando, entonces sudo lo interpreta como 'puede hacer uso del comando lsmod pero sin argumentos'. En este caso el usuario 'mariajose' podrá ver la lista de módulos del kernel, pero solo eso.**

Tags (etiquetas de comandos)

- Cuando se definen reglas, en la lista de comandos, estos pueden tener cero (como en los ejemplos anteriores) o más tags.
- Existen 6 de estas etiquetas o tags,

Tags: NOPASSWD Y PASSWD

Por defecto sudo requiere que cualquier usuario se identifique o autentifique con su contraseña.

- En la sección de 'Opciones' o 'Defaults' vimos que es posible indicar que un usuario o alias de usuario no requiera de autentificación.
- Pero el control granular propio de sudo, permite ir aun más lejos al indicar a nivel de comandos, cuáles requieren contraseña para su uso y cuáles no.

Tags: NOPASSWD Y PASSWD

  - gerardo webserver = NOPASSWD: /bin/kill, /usr/bin/lprm, /etc/httpd/conf/

- **Al usuario 'gerardo' en el equipo 'webserver' le indicamos que no requerirá contraseña para los comandos listados.**
- **El tag se hereda, es decir no solo el primer elemento de la lista de comandos, sino los subsiguientes.**
- **Suponiendo que el último '/etc/httpd/conf/' elemento, que permite modificar cualquier archivo contenido en el directorio, si deseamos que use contraseña, lo siguiente lo conseguirá:**
  - gerardo webserver = NOPASSWD: /bin/kill, /usr/bin/lprm, PASSWD: /etc/httpd/conf/
- **Aunque ya que solicitar contraseña es el default o defecto preestablecido, lo anterior también funcionará de la siguiente manera:**
  - gerardo webserver = /etc/httpd/conf/, NOPASSWD: /bin/kill, /usr/bin/lprm,

Tags: NOEXEC Y EXEC

- **Este es un tag muy importante a considerar cuando se otorgan permisos sobre programas que permiten escapes a shell (shell escape), como en el editor 'vi' que mediante el uso de '!' es posible ejecutar un comando en el shell sin salir de 'vi'.**
- **Con el tag NOEXEC se logra que esto no suceda, aunque no hay que tomarlo como un hecho, ya que siempre existe la posibilidad de vulnerabilidades no conocidas en los múltiples programas que utilizan escapes a shell.**
- **Al igual que los tags anteriores, el tag se hereda y se deshabilita con su tag contrario (EXEC), en caso de que en la lista de comandos hubiera varios comandos.**
  - clotilde ALL = NOEXEC: /usr/bin/vi

Tags: SETENV Y NOSETENV

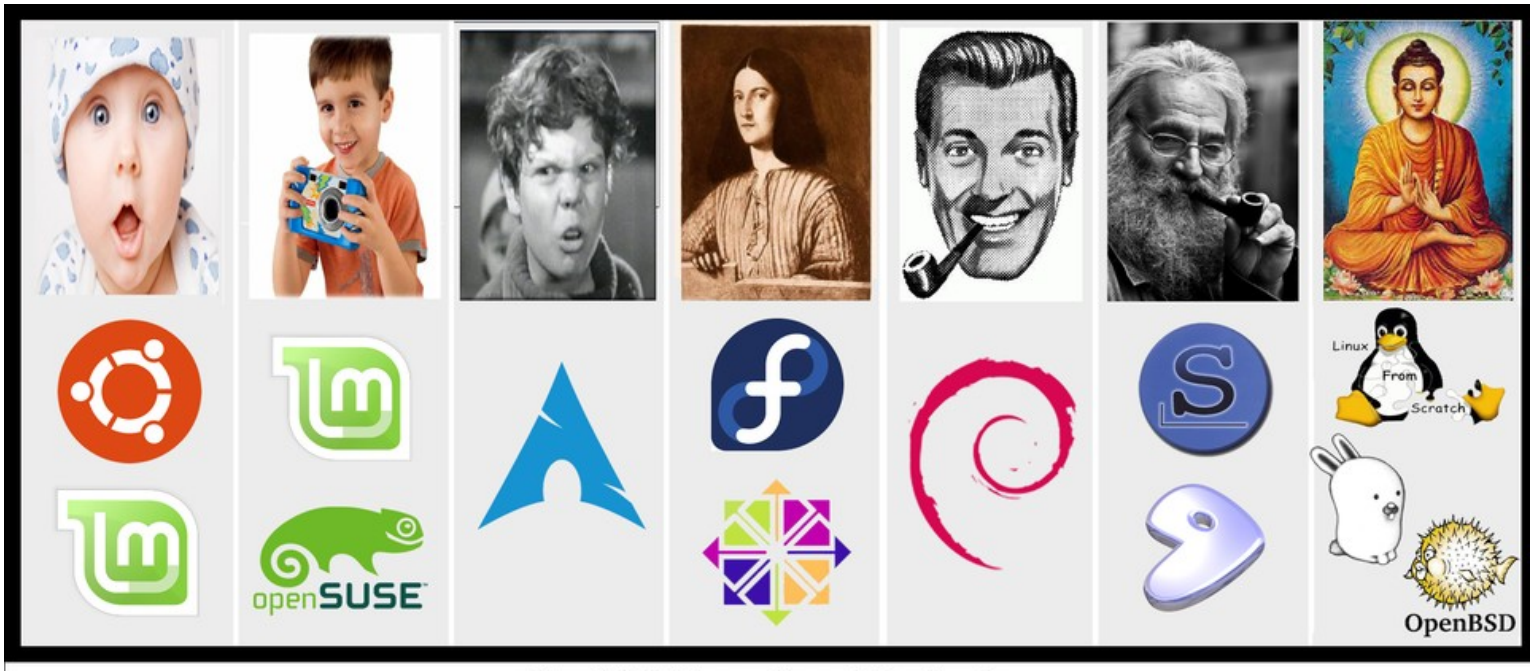
- 'setenv' por defecto para todos los usuarios esta establecida en 'off'.
- Esta opción si se activa por usuario (Defaults:sergio setenv) permitirá al usuario indicado cambiar el entorno de variables del usuario del cual tiene permisos de ejecutar comandos, y como generalmente este es 'root' resulta bastante peligrosa esta opción.
- A nivel de lista de comandos, es posible entonces especificar el tag 'SETENV' a un solo comando o a una pequeña lista de estos y solo cuando se ejecuten estos se podrán alterar su entorno de variables.
- Es decir, en vez de establecerlo por usuario, sería mas conveniente establecerlo por comando a ejecutarse solamente.

Tags: SETENV Y NOSETENV

  - ADMIN ALL = SETENV: /bin/date, NOSETENV ALL
- **A los usuarios definidos en el alias de usuario 'ADMIN' en cualquier host, pueden alterar las variables de entorno cuando ejecuten el comando 'date' (que puede ser útil por ejemplo para cambiar variables del tipo LOCALE), y cualquier otro comando, no tendrá esta opción al habilitar el tag contrario 'NOSETENV'.**
- **Y ya que este es el default, también sería válido de la siguiente manera y harían lo mismo:**
  - ADMIN ALL = ALL, SETENV: /bin/date.

Fin

- Soy la diapositiva número 66
- Fui concebida solamente para que la presentación no contara con 65 diapositivas
- Fin.



The GNU/Linux User Life Cycle