ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"DEFINICIÓN E IMPLEMENTACIÓN DE POLÍTICA DE SEGURIDAD DE USUARIOS EN LOS SERVIDORES DE SISTEMAS DE UNA ORGANIZACIÓN PROVEEDORA DE SERVICIOS DE DATOS E INTERNET."

EXAMEN DE GRADO (COMPLEXIVO)

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

Carlos Victoriano Calero Pérez
Guayaquil – Ecuador
2015

AGRADECIMIENTO

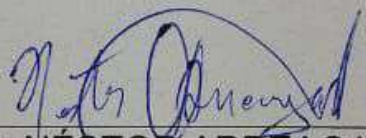
Agradezco a Dios por la vida y la salud, necesarias para poder realizar el presente trabajo. A mis padres, hermano, novia, familiares y amigos cercanos que siempre me animaron a seguir luchando y me apoyaron con sus oraciones. Dios los bendiga siempre.

DEDICATORIA

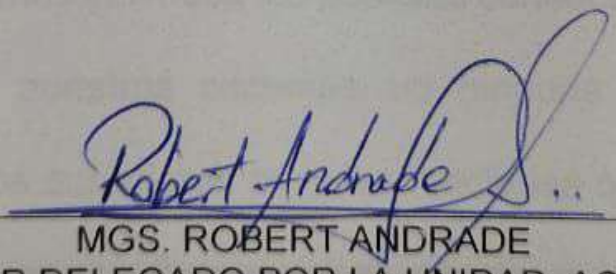
Dedico el presente trabajo a las personas verdaderamente cercanas a mí, con las cuales las alegrías se multiplican y las penas se dividen y se vencen: Victoriano, Gloria, Marcel, Mirella y Marilú. También a mis buenos amigos Manuel y Gerardo.

TRIBUNAL DE SUSTENTACIÓN

ING. LENÍN FREIRE
DIRECTOR DE LA MSIA



MGS. NÉSTOR ARREAGA
PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA



MGS. ROBERT ANDRADE
PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

RESUMEN

Debido a la necesidad, cada vez más creciente, de estar interconectados hemos enfocado nuestra atención a la seguridad necesaria al momento de usar las aplicaciones que nos permiten el acceso a nuestros datos, sin embargo hemos tenido la tendencia de manejar de una forma muy ligera la seguridad de los servidores/sistemas operativos que soportan a esas aplicaciones.

Esa actitud en muchas ocasiones nos ha llevado a perder el control de ciertas acciones y las ejecutamos sin medir las posibles consecuencias. Por ejemplo: usuarios creados en nuestros sistemas sin ninguna fecha de caducidad, usuarios con privilegios superiores para las funciones que van a realizar, etc. Estas son algunas de las prácticas que han creado un ambiente propicio para que amenazas, tanto internas como externas se aprovechen de nuestras debilidades y exploten nuestras malas prácticas logrando acceder a información sensible usándola de forma negativa.

El presente trabajo nos invita a definir Políticas de Seguridad en nuestros Sistemas de Cómputo y más que nada a aplicarlas en todos los niveles

jerárquicos creando conciencia de la necesidad de tenerla siempre presente en el trabajo diario.

De forma práctica se sugiere una política de seguridad para los usuarios y se detallan los pasos necesarios para aplicarla.

Para finalizar se mostrará con ejemplos las diferencias entre un sistema sin una política de seguridad de usuarios y otro sistema en el cual se haya aplicado la política sugerida.

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

El trabajo desarrollado a continuación nos mostrarán los problemas que tenía un departamento de sistemas de una empresa, describiendo el ambiente en el cual se administraban los servidores.

Se definió que los problemas detallados serían corregidos por medio de la aplicación de una Política de Seguridad de los Usuarios y para ello se definirán las condiciones que deberán cumplirse dentro de la política.

Con lo descrito en el primer capítulo se cubre el entorno teórico: definiciones, mientras en el segundo capítulo se detallarán las acciones específicas que permitirán la aplicación de la política definida.

En el tercer capítulo se muestran algunas diferencias al aplicar las acciones descritas en el capítulo 2, y se recomienda al lector realizar pruebas adicionales, si las considera necesarias, antes de aplicar en producción la política sugerida.

CAPÍTULO 1

GENERALIDADES

1.1. DESCRIPCIÓN DEL PROBLEMA

La organización proveedora de servicios de datos e internet en la cual vamos a laborar cuenta con un departamento de sistemas que se encarga del desarrollo de sistemas informáticos usando herramientas open source para uso de clientes internos y/o externos.

El departamento de sistemas además del desarrollo de software interno, eventualmente evalúa soluciones que no son open source antes de ser implementadas en la empresa. Los servidores donde se realiza el desarrollo de las soluciones y donde se ejecutan en producción cuentan con sistemas operativos Linux CentOS, principalmente versiones 5.x y 6.x.

Entre los inconvenientes que actualmente tiene la gestión de seguridad del departamento de sistemas podemos mencionar que:

- Todos los usuarios que acceden a los servidores, sean estos de desarrollo o producción, se les modifica su id de usuario para que sean root, de tal forma que puedan ejecutar todos los comandos requeridos para su tarea.

- Los usuarios conforme requieren nuevos paquetes para sus desarrollos los instalan bajo su personal criterio.

- Si los usuarios requieren actualizaciones en los equipos de producción, cualquier de ellos está en capacidad de realizar las modificaciones sin tener un control adecuado de las mismas.

- El departamento de seguridad interna de la organización requiere eventualmente que se actualicen paquetes que están propensos a vulnerabilidades y esta tarea también puede ser realizada por cualquier usuario, generalmente alguno de los jefes que esté más libre de otras tareas.

- Las claves usadas por los usuarios son débiles y se actualizan a discreción personal.

- Al preparar un nuevo servidor la selección del tipo de instalación se realiza de acuerdo al criterio del encargado de ese nuevo equipo.

- Equipos de desarrollo se convierten en equipos de producción sin un adecuado proceso de aseguramiento.

- El número aproximado de servidores en este escenario es de 80.

1.2. SOLUCIÓN PROPUESTA

Con la definición de la Política Seguridad de los Usuarios [1] de los servidores de sistemas se prevé los siguientes beneficios:

- Tener un control adecuado sobre el usuario administrador del sistema operativo: root.

- Registrar las modificaciones realizadas por los usuarios en los registros de seguridad respectivos y poder reconocer a los usuarios y los comandos ejecutados por ellos.

- Restringir las tareas de actualización o modificación de archivos de configuración a los usuarios para los cuales se les haya dado autorización para realizarlas.

- Instalar en servidores nuevos el menor número de paquetes innecesarios, endureciendo desde el inicio la seguridad de los equipos.

- Realizar una depuración general de los servicios y paquetes innecesarios en los servidores que pasan de desarrollo a producción.

CAPÍTULO 2

METODOLOGÍA DE DESARROLLO DE LA SOLUCIÓN

Tomando en cuenta la descripción del problema que enumera los principales problemas de la situación actual sobre la gestión de usuarios en los servidores administrados por el departamento de sistemas, se definió la política de seguridad que se detallará a continuación.

2.1. Política de Seguridad de los Usuarios de los Servidores del Departamento de Sistemas – Sistema Operativo: Linux

Generalidades

1. La presente política será aplicada exclusivamente en los servidores del departamento de Sistemas
2. Se enfocará en los servidores con sistema operativo Linux. Al momento de escribirse esta política se cuenta con CentOS 5.x, 6.x y 7.x.

Asignación de usuarios

1. Los usuarios serán asignados a los colaboradores del departamento de Sistemas.
2. Los usuarios se otorgarán previa autorización de los jefes de área o de la gerencia de sistemas.
3. De ser necesario se otorgarán usuarios a colaboradores de otros departamentos, previa solicitud del jefe del departamento que requiere el acceso y previa autorización de los jefes de área o de la gerencia de sistemas.
4. Se otorgarán usuarios nombrados, y se crearán grupos de acuerdo a su rol en los servidores. Si los usuarios pertenecen a un departamento externo se creará un nuevo grupo.
5. Se crearán usuarios para ser usadas por las aplicaciones, cuando estas requieran acceso remoto desde otros servidores. La creación de usuarios para las aplicaciones deberá ser autorizada por la gerencia de sistemas.

Asignación del usuario *root*

1. El único autorizado para usar el usuario root de los servidores será el Administrador de Infraestructura, o la persona(s) que realice(n) sus funciones.

2. La gerencia de sistemas tendrá un usuario en los servidores y deberá tener permisos de administrarlos como lo haría el usuario root.

Claves

1. La clave de los usuarios deberá caducar cada 90 días.
2. La clave de los usuarios deberá contener mínimo una minúscula, una mayúscula, un dígito y un caracter especial.
3. La clave de los usuarios deberá tener una longitud mínima de 10 caracteres.
4. Cuando el sistema pida actualizar la clave por caducidad, esta no podrá ser ninguna de las últimas 10 claves usadas.
5. Las claves de los usuarios de aplicaciones no deben tener fecha de caducidad.

Accesos Remotos SSH

1. Por defecto el usuario ***root*** no podrá acceder remotamente a los servidores usando un cliente SSH.
2. El acceso del usuario ***root*** a los servidores estará restringido a direcciones IP bien conocidas.

3. Entre las direcciones IP bien conocidas debe constar la IP(s) que use el Administrador de Infraestructura.

Permisos

1. Por defecto todos los usuarios nuevos se crearán como usuarios del grupo ***operadores***, que no tendrá definido ningún permiso especial.

2. El directorio personal de los nuevos usuarios será el que por defecto le asigne el sistema operativo.

3. Todos los usuarios, excepto ***root*** deben contar con reglas de sudo para la ejecución de comandos que pudieran tener incidencia en la configuración del sistema operativo y/o las aplicaciones que se ejecutan en él.

4. De forma independiente, por cada servidor, deben elegirse cuáles son los comandos/aplicaciones que serán ejecutados por los usuarios y con qué ID de usuario los ejecutarán.

5. En casos excepcionales, cuando no haya otra forma de hacerlo, se asignará la propiedad de carpetas específicas a usuarios o al grupo que contenga los usuarios que las gestionarán. Por ejemplo los miembros del personal de Calidad son los que, dentro de sus funciones, deben actualizar las nuevas versiones de los

programas desarrollados internamente; en ese caso la carpeta donde residen los programas será asignada al grupo creado para esta función.

2.2. Aplicación Práctica de la política definida

En la aplicación de la política definida se usarán principalmente archivos de configuración de PAM [2].

2.2.1. Creación de grupos personalizados

Se definen los siguientes grupos básicos: operadores, administradores, dba y sqa. Para crear los grupos acordados ejecutar los siguientes comandos [3]:

groupadd -g 10000 operadores

groupadd -g 10001 administradores

groupadd -g 10002 dba

groupadd -g 10003 sqa

2.2.2. Definición de valores por defecto

Se modificarán archivos que contienen las configuraciones de los valores por defecto de:

- La caducidad de las claves [3]

- El grupo predeterminado de los nuevos usuarios.

Para modificar el tiempo de caducidad de las claves editar el archivo **/etc/login.defs** y modificar la línea **PASS_MAX_DAYS 99999** por **PASS_MAX_DAYS 90**.

Para modificar el grupo predeterminado de los nuevos usuarios:

- editar el archivo **/etc/default/useradd** y modificar la línea **GROUP=100** por **GROUP=10000**, y

- editar el archivo /etc/login.defs y modificar la línea USERGROUPS_ENAB yes por USERGROUPS_ENAB no.

2.2.3. Creación de usuarios

Para crear un nuevo usuario que no tendrá ningún permiso especial ejecutar cualquiera de los siguientes comandos:

useradd -g 10000 -c "Juan Pueblo" jpueblo

useradd -c "Juan Pueblo" jpueblo

Dado que el grupo por defecto es el de los operadores no es necesario especificarlo al crear el usuario.

En cambio para crear un nuevo usuario que pertenecerá al grupo **administradores** ejecutar el siguiente comando: useradd -g 10001 -c "Carlos Calero" ccalero

2.2.4. Claves

2.2.4.1. Generación y Asignación de Clave Fuerte

Para generar un clave complicada de 20 caracteres ejecutar el siguiente comando:

tr -dc A-Za-z0-9A-Za-z0-9%\&\(\)*_[]{}- < /dev/urandom | head -c 20 | xargs

Para modificar la clave de un usuario y asignarle una clave complicada copiar la clave generada anteriormente y aplicarla al comando siguiente cuando solicite la nueva clave:

passwd ccalero

2.2.4.2. Restricciones de fortaleza de la clave

Para modificar la fortaleza de las claves de los usuarios editar el archivo **/etc/pam.d/system-auth**:

En versión 5.x modificar las siguientes líneas:

password requisite pam_cracklib.so try_first_pass retry=3

password sufficient pam_unix.so md5 shadow nullok try_first_pass use_authtok

para que sean como las siguientes:

password requisite pam_cracklib.so retry=3 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1

password requisite pam_unix.so sha512 shadow nullok use_authtok remember=10 minlen=10

En versión 6.x modificar las siguientes líneas:

password requisite pam_cracklib.so try_first_pass retry=3 type=

password sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok

para que sean como las siguientes:

password requisite pam_cracklib.so retry=3 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1

password requisite pam_unix.so sha512 shadow nullok use_authtok remember=10 minlen=10

En versión 7.x modificar las siguientes líneas:

password requisite pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=

password sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok

para que sean como las siguientes:

password requisite pam_pwquality.so retry=3 lcredit=-1 ucredit=-1 dcredit=-1 ocredit=-1

password requisite pam_unix.so sha512 shadow nullok try_first_pass use_authtok remember=10 minlen=10

2.2.4.3. Opciones del fortalecimiento de claves

Las modificaciones realizadas en el punto anterior fortalecen las claves usadas por los usuarios cuando ellos decidan modificar sus claves:

- lcredit=1: verifica que la clave contenga al menos una letra minúscula.
- ucredit=1: verifica que la clave contenga al menos una letra mayúscula.
- dcredit=1: verifica que la clave contenga al menos un dígito.
- ocredit=1: verifica que la clave contenga al menos un carácter de otro tipo.
- remember=10: verifica que la clave ingresada no sea igual a ninguna de las 10 últimas claves anteriores.
- minlen=10: verifica que la clave tenga al menos 10 caracteres.

2.2.4.4. Modificando caducidad de claves de aplicaciones

Asumir que la **aplicación 1** tiene un usuario en nuestro servidor y la cuenta creada es **appuser01**. Antes de modificar la fecha de caducidad de la clave de este usuario confirmar con el comando siguiente los valore originales:

chage -l appuser01

La salida del comando anterior debe ser similar a las siguientes líneas:

Last password change : Aug 21, 2014

Password expires : Nov 19, 2014

Password inactive : never

Account expires : never

Minimum number of days between password change : 0

Maximum number of days between password change : 90

Number of days of warning before password expires : 7

El valor de la segunda línea: Password expires, indica la fecha en la cual expira la clave del usuario **appuser01** y esta fecha se debe a la penúltima línea que indica cuantos días la clave es válida. Por lo tanto para que no exista fecha de caducidad se debe ejecutar el siguiente comando:

chage -M -1 appuser01

El comando anterior remueve la verificación de la fecha de caducidad. Verificar nuevamente con el comando siguiente:

chage -l appuser01

Notar la diferencia de la salida siguiente con respecto a la salida inicial:

Last password change : Aug 21, 2014

Password expires : never

Password inactive : never

Account expires : never

Minimum number of days between password change : 0

Maximum number of days between password change : -1

Number of days of warning before password expires : 7

2.2.5. Restricción de accesos remoto del usuario root

La restricción que será tratada en este punto es independiente de las restricciones generales de acceso definidas usando el firewall del sistema operativo o el archivo **/etc/hosts.allow**, ya que esas restricciones se aplican independientemente del usuario que se esté conectando.

Nuestra meta, por ahora, es restringir solamente el acceso del usuario **root** y no el de todos los usuarios y para ello debemos editar dos archivos, en el primero indicaremos las reglas que se usarán para permitir/negar el acceso y en el segundo especificaremos que se use el archivo de las definiciones declaradas en el primero.

El primer archivo en editar es **/etc/security/access.conf** [3]. Por defecto todas las líneas de este archivo están comentadas explicando su uso. Agregar las siguientes líneas al final del archivo:

+ : administradores : ALL

- : operadores : 127.0.0.1

+ : operadores : ALL

+ : root : cron crond :0 tty1 tty2 tty3 tty4 tty5 tty6

+ : root : 127.0.0.1

+ : root : 172.123.25.111 172.123.26.111 172.123.25.1

- : root : ALL

Las líneas anteriores especifican las siguientes reglas de acceso:

- Permitir el acceso a los usuarios del grupo **administradores** desde todas las IPs.
- Negar el acceso a los usuarios del grupo **operadores** desde **localhost**.
- Permitir el acceso a los usuarios del grupo **operadores** desde todas las IPs.
- Permitir el acceso del usuario **root** via **cron** y **crond** desde la consola y las terminales especificadas.
- Permitir el acceso del usuario **root** desde **localhost**.

- Permitir el acceso del usuario **root** desde IPs bien conocidas, que permitirán el acceso de forma centralizada y desde la IP del administrador de infraestructura.

- Negar el acceso al usuario **root** desde todas las IPs

Las líneas son “ejecutadas” en el orden en que aparecen y la verificación se detiene cuando una regla se cumple.

El segundo archivo a editar es **/etc/pam.d/sshd**. En ese archivo se debe agregar la línea siguiente:

account required pam_access.so

La línea agregada obliga a tomar en cuenta el archivo **/etc/security/access.conf** para toda conexión **ssh** que se intente hacia nuestro servidor. Importante ubicarla antes de cualquier línea que empiece con **account** y después de la(s) línea(s) que empiecen con **auth**.

2.2.6. Restricción de sustitución de usuario root (su)

Para restringir que sólo los usuarios del grupo **administradores** puedan sustituir al usuario **root** se debe editar el archivo **/etc/pam.d/su** [4] y se debe modificar la línea:

#auth required pam_wheel.so use_uid

Por la siguiente línea:

auth required pam_wheel.so use_uid group=administradores

Con la configuración anterior se consigue que sólo un usuario del grupo **administradores** pueda ejecutar el comando **su** para poder usar el usuario **root**.

2.2.7. Ejecución de comandos como usuario root (sudo)

El programa **sudo** (**S**ubstitute **U**ser **DO**) [5] permite que un usuario ejecute comandos con los privilegios que tiene un segundo usuario. Generalmente el segundo usuario es el usuario ***root***.

Cuando se lo usa, por defecto, el sistema solicita la clave del primer usuario previo a la ejecución solicitada.

En nuestro caso, dado que la política especifica que se permita la ejecución de comandos como si fueran ejecutado por el usuario ***root***, definiremos reglas para **sudo**.

Para ´la definición de las reglas usaremos el editor **visudo**, el cual antes de grabar verificará que las configuraciones este correctamente escritas y nos alertará en caso de encontrar algún error en la sintaxis.

Para escribir reglas se recomienda definir previamente alias sobre los que haremos referencia en las reglas. Los alias disponibles son los siguientes:

- **Cmnd_Alias**: Para definir lista de programas que se permitirá o negará su ejecución.
- **User_Alias**: Para definir lista de usuarios que se les permitirá o se les negará el uso de sudo.
- **Host_Alias**: Para definir lista de servidores desde los cuales se permitirá o negará el uso de sudo.
- **Runas_Alias**: Para definir lista de identidades de usuarios que serán permitidos o prohibidas de usar sudo.

En nuestro caso, dada la variedad de servidores administrados por el Departamento de Sistemas se ve necesario realizar varias

plantillas, que se podrían agrupar en tres grupos principales de servidores:

- **de Infraestructura**: Estos serán los servidores que proveen servicios que son usados por los demás servidores, por ejemplo: ldap, sso, etc.

- **de Base de Datos**: Estos serán los servidores que tienen instalados servicios de bases de datos, por ejemplo mysql, Oracle, etc.

- **de Aplicaciones**: Estos servidores son los que tiene aplicaciones específicas, como por ejemplo Aplicación Financiera, Aplicación de Soporte, Aplicación de Operaciones, etc.

Los siguientes puntos detallarán las plantillas básicas generadas para los grupos definidos anteriormente. Sólo se mencionarán las líneas que deben agregarse al final del archivo que se abra al ejecutar el comando **visudo**.

2.2.7.1. Plantilla Básica para Servidores de Infraestructura

```
Cmnd_Alias ALMACENAMIENTO = /bin/df, /bin/du, /bin/mount, \
```

```
                /bin/umount
Cmnd_Alias ARCHIVOS = /usr/bin/diff, /bin/find, /bin/ls, \
                /usr/bin/updatedb, /bin/cp, /bin/mkdir, /bin/mv, \
                /bin/rm, /bin/tar
Cmnd_Alias EDITORES= /bin/cat, /bin/grep, /usr/bin/head, \
                /usr/bin/less, /bin/more, /usr/bin/tail, /bin/nano, \
                /bin/vi, /usr/bin/vim
Cmnd_Alias ENERGIA= /sbin/poweroff, /sbin/reboot
Cmnd_Alias NEGADOS_01 = !/bin/nano passwd, !/bin/nano group, \
                !/bin/nano shadow, !/bin/nano hosts.allow, \
                !/bin/nano hosts.deny, !/bin/vi passwd, \
                !/bin/vi group, !/bin/vi shadow, !/bin/vi hosts.allow, \
                !/bin/vi hosts.deny, !/usr/bin/vim passwd, \
                !/usr/bin/vim group, !/usr/bin/vim shadow, \
                !/usr/bin/vim hosts.allow, !/usr/bin/vim hosts.deny
Cmnd_Alias NEGADOS_02 = !/bin/nano /etc/passwd, \
                !/bin/nano /etc/group, !/bin/nano /etc/shadow, \
                !/bin/nano /etc/hosts.allow, !/bin/nano \
                /etc/hosts.deny, !/bin/vi /etc/passwd, \
                !/bin/vi /etc/group, !/bin/vi /etc/shadow, \
                !/bin/vi /etc/hosts.allow, !/bin/vi /etc/hosts.deny, \
                !/usr/bin/vim /etc/passwd, !/usr/bin/vim /etc/group,
                !/usr/bin/vim /etc/shadow,
```

```
                !/usr/bin/vim /etc/hosts.allow, \
                !/usr/bin/vim /etc/hosts.deny
Cmnd_Alias PERMISOS      = /usr/bin/passwd, /bin/chgrp, \
                /bin/chmod, /bin/chown
Cmnd_Alias PROCESOS      = /usr/bin/kill, /usr/bin/killall, /bin/nice, \
                /bin/ps, /usr/bin/top
Cmnd_Alias REDES   = /sbin/ip addr, /sbin/mii-tool, /bin/netstat, \
                /bin/ping, sbin/ifconfig, /sbin/route
Cmnd_Alias SEGURIDAD     = /sbin/iptables
Cmnd_Alias SERVICIOS      = /sbin/chkconfig, /usr/bin/crontab, \
                /sbin/service
Cmnd_Alias SOFTWARE      = /bin/rpm, /usr/bin/yum
User_Alias ADMINISTRADORES = %administradores
ADMINISTRADORES ALL      = (root) ALMACENAMIENTO, ARCHIVOS, \
                EDITORES, ENERGIA, PROCESOS, REDES, \
                NEGADOS_01, NEGADOS_02
ccalero ALL      = (root) PERMISOS, SEGURIDAD, \
                SERVICIOS, SOFTWARE
jmsuarez ALL  = (root) PERMISOS, SEGURIDAD, SOFTWARE
```

2.2.7.2. Plantilla Básica para Servidores de Base de Datos

```
Cmnd_Alias ALMACENAMIENTO = /bin/df, /bin/du, /bin/mount, \
```

```
                /bin/umount
Cmnd_Alias ARCHIVOS     = /usr/bin/diff, /bin/find, /bin/ls, \
                /usr/bin/updatedb, /bin/cp, /bin/mkdir, /bin/mv,
                /bin/rm, /bin/tar
Cmnd_Alias EDITORES     = /bin/cat, /bin/grep, /usr/bin/head, \
                /usr/bin/less, /bin/more, /usr/bin/tail, /bin/nano, \
                /bin/vi, /usr/bin/vim
Cmnd_Alias ENERGIA      = /sbin/poweroff, /sbin/reboot
Cmnd_Alias NEGADOS_01 = !/bin/nano passwd, !/bin/nano group, \
                !/bin/nano shadow, !/bin/nano hosts.allow, \
                !/bin/nano hosts.deny, !/bin/vi passwd, \
                !/bin/vi group, !/bin/vi shadow, !/bin/vi hosts.allow, \
                !/bin/vi hosts.deny, !/usr/bin/vim passwd, \
                !/usr/bin/vim group, !/usr/bin/vim \
                shadow, !/usr/bin/vim hosts.allow, \
                !/usr/bin/vim hosts.deny
Cmnd_Alias NEGADOS_02 = !/bin/nano /etc/passwd, !/bin/nano \
                /etc/group, !/bin/nano /etc/shadow, \!/bin/nano \
                /etc/hosts.allow, !/bin/nano /etc/hosts.deny, \
                !/bin/vi /etc/passwd, !/bin/vi /etc/group, \
                !/bin/vi /etc/shadow, \
                !/bin/vi /etc/hosts.allow, !/bin/vi /etc/hosts.deny, \
                !/usr/bin/vim /etc/passwd, \
```

```
                !/usr/bin/vim /etc/group,
                !/usr/bin/vim /etc/shadow, \
                !/usr/bin/vim /etc/hosts.allow, \
                !/usr/bin/vim /etc/hosts.deny
Cmnd_Alias PERMISOS      = /usr/bin/passwd, /bin/chgrp, \
                /bin/chmod, /bin/chown
Cmnd_Alias PROCESOS      = /usr/bin/kill, /usr/bin/killall, /bin/nice, \
                /bin/ps, /usr/bin/top
Cmnd_Alias REDES   = /sbin/ip addr, /sbin/mii-tool, /bin/netstat, \
                /bin/ping, /sbin/ifconfig, /sbin/route
Cmnd_Alias SEGURIDAD     = /sbin/iptables
Cmnd_Alias SERVICIOS      = /sbin/chkconfig, /usr/bin/crontab, \
                /sbin/service
Cmnd_Alias SERVICIOS_DBA = /sbin/chkconfig, /usr/bin/crontab, \
                /sbin/service mysql start, /sbin/service mysql stop, \
                /sbin/service mysql restart
Cmnd_Alias SOFTWARE      = /bin/rpm, /usr/bin/yum
User_Alias ADMINISTRADORES = %administradores
User_Alias DBA = %dba
ADMINISTRADORES ALL      = (root) ALMACENAMIENTO, ARCHIVOS, \
                EDITORES, ENERGIA, PERMISOS, PROCESOS, \
                REDES, SEGURIDAD, SERVICIOS, \
                SOFTWARE, NEGADOS_01, NEGADOS_02
```

```
DBA ALL         = (root) ALMACENAMIENTO, ARCHIVOS, \
                  EDITORES, ENERGIA, PROCESOS, \
                  SERVICIOS_DBA, NEGADOS_01, NEGADOS_02
ccalero ALL     = (root) PERMISOS, SEGURIDAD, SOFTWARE
jmsuarez ALL  = (root) PERMISOS, SEGURIDAD, SOFTWARE
```

2.2.7.3. Plantilla Básica para Servidores de Aplicaciones

```
Cmnd_Alias ALMACENAMIENTO = /bin/df, /bin/du, /bin/mount, \
                  /bin/umount
Cmnd_Alias ARCHIVOS = /usr/bin/diff, /bin/find, /bin/ls, \
                  /usr/bin/updatedb, \
                  /bin/cp, /bin/mkdir, /bin/mv, /bin/rm, /bin/tar
Cmnd_Alias EDITORES       = /bin/cat, /bin/grep, /usr/bin/head, \
                  /usr/bin/less, /bin/more, /usr/bin/tail, /bin/nano, \
                  /bin/vi, /usr/bin/vim
Cmnd_Alias ENERGIA        = /sbin/poweroff, /sbin/reboot
Cmnd_Alias NEGADOS_01 = !/bin/nano passwd, !/bin/nano group, \
                  !/bin/nano shadow, !/bin/nano hosts.allow, \
                  !/bin/nano hosts.deny, !/bin/vi passwd, \
                  !/bin/vi group, !/bin/vi shadow, !/bin/vi hosts.allow, \
                  !/bin/vi hosts.deny, !/usr/bin/vim passwd, \
                  !/usr/bin/vim group, !/usr/bin/vim shadow, \
```

```
                !/usr/bin/vim hosts.allow, !/usr/bin/vim hosts.deny
Cmnd_Alias NEGADOS_02 = !/bin/nano /etc/passwd, !/bin/nano \
                /etc/group, !/bin/nano /etc/shadow, \
                !/bin/nano /etc/hosts.allow, \
                !/bin/nano /etc/hosts.deny, \
                !/bin/vi /etc/passwd, !/bin/vi /etc/group, \
                !/bin/vi /etc/shadow, !/bin/vi /etc/hosts.allow, \
                !/bin/vi /etc/hosts.deny, !/usr/bin/vim /etc/passwd, \
                !/usr/bin/vim /etc/group, \
                !/usr/bin/vim /etc/shadow, \
                !/usr/bin/vim /etc/hosts.allow,
                !/usr/bin/vim /etc/hosts.deny
Cmnd_Alias PERMISOS      = /usr/bin/passwd, /bin/chgrp, \
                /bin/chmod, /bin/chown
Cmnd_Alias PROCESOS      = /usr/bin/kill, /usr/bin/killall, /bin/nice, \
                /bin/ps, /usr/bin/top
Cmnd_Alias REDES   = /sbin/ip addr, /sbin/mii-tool, \
                /bin/netstat, /bin/ping, /sbin/ifconfig, /sbin/route
Cmnd_Alias SEGURIDAD     = /sbin/iptables
Cmnd_Alias SERVICIOS      = /sbin/chkconfig, /usr/bin/crontab, \
                /sbin/service
Cmnd_Alias SERVICIOS_SQA = /sbin/chkconfig, /usr/bin/crontab, \
                /sbin/service httpd start, /sbin/service httpd stop, \
```

```
                /sbin/service httpd restart
Cmnd_Alias SOFTWARE     = /bin/rpm, /usr/bin/yum
User_Alias ADMINISTRADORES = %administradores
User_Alias SQA = %sqa
ADMINISTRADORES ALL     = (root) ALMACENAMIENTO, ARCHIVOS, \
                EDITORES, ENERGIA, PERMISOS, PROCESOS, \
                REDES, SEGURIDAD, SERVICIOS, \
                SOFTWARE, NEGADOS_01, NEGADOS_02
SQA ALL         = (root) ALMACENAMIENTO, ARCHIVOS, EDITORES, \
                ENERGIA, PROCESOS, SERVICIOS_SQA, SOFTWARE, \
                NEGADOS_01, NEGADOS_02
ccalero ALL             = (root) PERMISOS, SEGURIDAD, SOFTWARE
jmsuarez ALL            = (root) PERMISOS, SEGURIDAD, SOFTWARE
```

CAPÍTULO 3

ANÁLISIS DE RESULTADOS

3.1. Ambiente de Demostración

3.1.1. Arquitectura de Servidores

Nuestro ambiente de pruebas será virtualizado y contará con 4 servidores, todos con CentOS 6.x, instalación mínima del sistema operativo, 512Mb de RAM, disco duro de 32Gb, 2 CPUs.

Los servidores serán configurados con las siguientes direcciones IP:

Tabla 3.1. Lista de servidores del ambiente de demostración.

#	Hostname	Dirección IP
1	accesocentral1	172.123.25.111
2	sinacceso	172.123.25.222
3	servidor1	172.123.25.11
4	servidor2	172.123.25.12

La figura siguiente ilustra la arquitectura de nuestro ambiente de demostración:

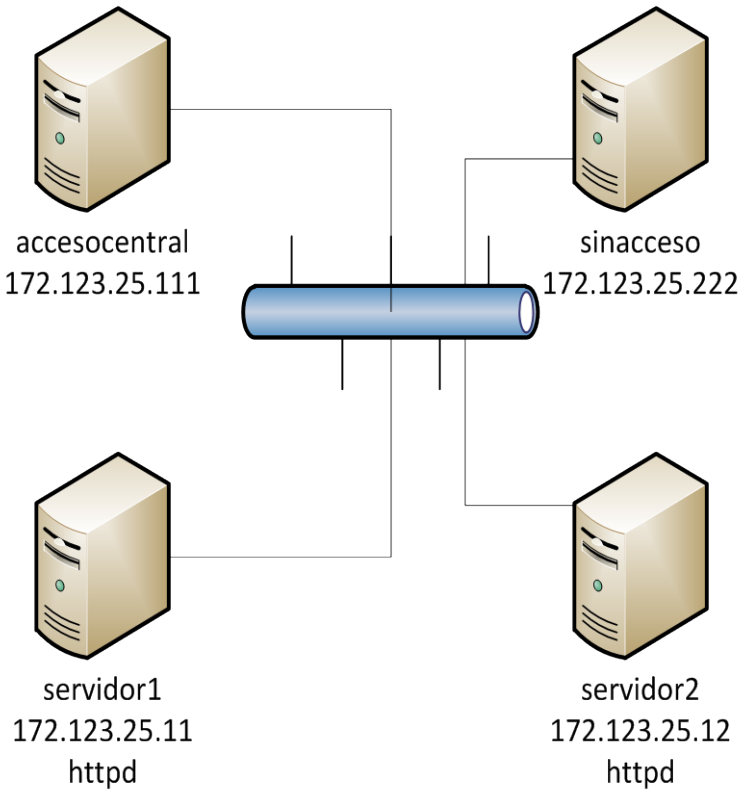


Figura 3.1. Arquitectura de ambiente de pruebas.

3.1.2. Condiciones adicionales de los servidores

Los servidores mencionados anteriormente deberán cumplir las siguientes condiciones adicionales respecto a los permisos que tendrán entre ellos.

- **accesocentral1**: Servidor desde el cual se tendrá acceso como **root** a los servidores 1 y 2. No tendrá ningún paquete adicional a la instalación mínima.

- **sinacceso**: Servidor desde el cual no se tendrá acceso con el usuario **root** a los servidores 1 y 2. No tendrá ningún paquete adicional a la instalación mínima.

- **servidor1**: Servidor en el cual se mostrará el estado actual del departamento de sistemas, sin aplicación de ninguna política de seguridad de usuarios.

- **servidor2**: Servidor en el cual se aplicará la política de seguridad de usuarios.

- Los servidores 1 y 2 tendrán instalado el servicio httpd

3.1.3. Usuarios

En los servidores 1 y 2 se crearán 4 usuarios, en el 1 sin aplicar política de seguridad de usuarios, en el 2 aplicándola. Los usuarios serán:

- **ccalero**: Usuario administrador de la infraestructura.

- **jmsuarez**: Gerente de Sistemas. Tendrá un usuario con privilegios como el administrador de infraestructura.

- **jpueblo**: Usuario sin privilegios de ninguna clase.

- **mleon**: Usuario del grupo administradores que no es administrador de la infraestructura.

En **servidor1** a los usuarios **ccalero**, **jmsuarez** y **mleon** se les modificará manualmente el **id** de usuario para que sean **root**. Los usuarios en ambos servidores tendrán como clave predeterminada la siguiente: **password.1**

3.2. Resultados de la aplicación de la Política de Seguridad de Usuarios

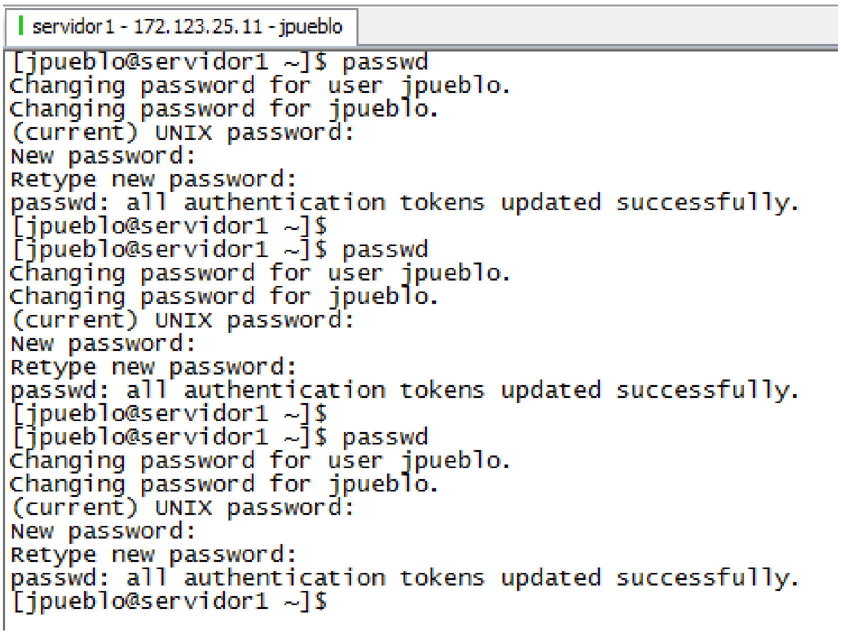
En este punto se mostrarán 4 ejemplos que nos permitirán observar diferencias entre un sistema con políticas y otro sin ellas.

3.2.1. Fortaleza de claves

Para esta demostración se intentará modificar la clave de un usuario proponiendo que las nuevas claves no cumplan con los requerimientos definidos en la política, bajo el título de claves y su definición en la política.

En la primera imagen se muestran los intentos de modificación de clave del usuario **jpueblo** en el **servidor1** en el siguiente orden:

- De clave inicial a **QWEZXCasd123**
- De QWEZXCasd123 a rtyvbnFGH
- De rtyvbnFGH a QWEZXCasd123



servidor1 - 172.123.25.11 - jpueblo

```
[jpueblo@servidor1 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[jpueblo@servidor1 ~]$
```

Figura 3.2. Pruebas de modificación de clave de usuario jpueblo en servidor1

En la segunda imagen se muestran los intentos de modificación de clave del usuario **jpueblo** en el **servidor2** en el siguiente orden:

- De clave inicial a QWEZXCasd123
    - Dado que la clave no es fuerte, probar con la clave QWEZXCasd!23
- De QWEZXCasd!23 a rtyvbnFGH567
    - Dado que la clave no es fuerte, probar con la clave **rtyvbnFGH%67**
- De rtyvbnFGH%67 a 45+vbnFGH
    - Dado que la clave no es fuerte, probar con la clave **QWEZXCasd!23**
- Dado que se intentó reusar una de las últimas 10 claves, probar el cambio con la clave **MNBjhgUYT32!**

```
[jpueblo@servidor2 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
BAD PASSWORD: is too simple
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
BAD PASSWORD: is too simple
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
BAD PASSWORD: is too simple
New password:
Retype new password:
Password has been already used. Choose another.
passwd: Authentication token manipulation error
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ passwd
Changing password for user jpueblo.
Changing password for jpueblo.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[jpueblo@servidor2 ~]$
```

Figura 3.3. Pruebas de modificación de clave de usuario jpueblo en servidor2

3.2.2. Acceso remoto a servidores

Para esta demostración se intentarán dos pruebas: la primera acceder con el usuario **root** a **servidor1** y **servidor2** y la segunda acceder con otros usuarios. Recordar el título de accesos remotos ssh y su definición en la política.

3.2.2.1. Usuario root

En la primera imagen se muestran los intentos de conexión de root desde el servidor 172.123.25.111 a los servidores 1 y 2 sin novedades. Este equipo tiene permiso en el servidor2 para que el usuario root se conecte remotamente de acuerdo a las políticas implementadas.

```
[root@accesocentral1 ~]# ssh -l root 172.123.25.11
root@172.123.25.11's password:
Last login: Mon Jul 20 21:34:54 2015 from 172.123.25.111
[root@servidor1 ~]#
[root@servidor1 ~]# exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
[root@accesocentral1 ~]# ssh -l root 172.123.25.12
root@172.123.25.12's password:
Last login: Mon Jul 20 21:35:33 2015 from 172.123.25.111
[root@servidor2 ~]#
[root@servidor2 ~]# exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.4. Intento de conexión de usuario root desde servidor permitido

En la segunda imagen se muestran los intentos de conexión de **root** desde el servidor 172.123.25.222. Se accede al **servidor1** sin novedades, sin embargo al **servidor2** no se tiene acceso, dado que este equipo no está permitido para que el usuario **root** se conecte remotamente de acuerdo a las políticas implementadas.

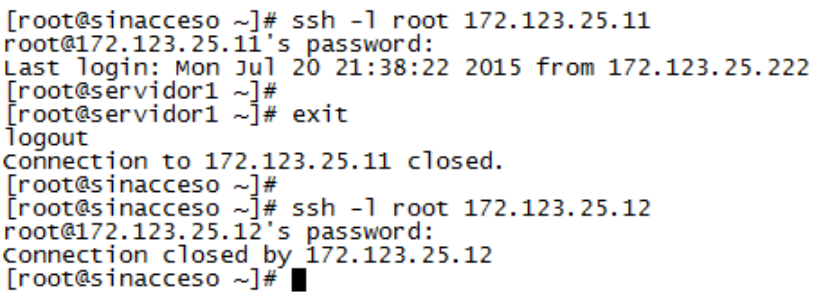


Figura 3.5. Intento de conexión de usuario root desde servidor no permitido

3.2.2.2. Otros usuarios

En la primera imagen se muestran los intentos de conexión de **jpueblo** y **ccalero** desde el servidor 172.123.25.111, el cual se logra sin novedades tanto al **servidor1** como al **servidor2**. Debemos notar que el usuario **ccalero** en el **servidor1** tiene **uid** de **root**.

```
[root@accesocentral1 ~]# ssh -l jpueblo 172.123.25.11
jpueblo@172.123.25.11's password:
Last login: Mon Jul 20 22:01:23 2015 from 172.123.25.222
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
[root@accesocentral1 ~]# ssh -l ccalero 172.123.25.11
ccalero@172.123.25.11's password:
Last login: Mon Jul 20 22:01:44 2015 from 172.123.25.222
[root@servidor1 ~]#
[root@servidor1 ~]# exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
[root@accesocentral1 ~]#
[root@accesocentral1 ~]#
[root@accesocentral1 ~]# ssh -l jpueblo 172.123.25.12
jpueblo@172.123.25.12's password:
Last login: Mon Jul 20 22:02:07 2015 from 172.123.25.222
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
[root@accesocentral1 ~]# ssh -l ccalero 172.123.25.12
ccalero@172.123.25.12's password:
Last login: Mon Jul 20 22:02:28 2015 from 172.123.25.222
[ccalero@servidor2 ~]$
[ccalero@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.6. Intento de conexión de usuarios jpueblo y ccalero desde servidor permitido

En la segunda imagen se muestran los intentos de conexión de **jpueblo** y **ccalero** desde el servidor 172.123.25.111, el cual se logra sin novedades tanto al **servidor1** como al **servidor2**. Se confirma de esta forma que la política no afecta el acceso de los usuarios que tienen un **uid** diferente de **root**.

```
[root@sinacceso ~]# ssh -l jpueblo 172.123.25.11
jpueblo@172.123.25.11's password:
Last login: Mon Jul 20 21:59:01 2015 from 172.123.25.222
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ exit
logout
Connection to 172.123.25.11 closed.
[root@sinacceso ~]#
[root@sinacceso ~]# ssh -l ccalero 172.123.25.11
ccalero@172.123.25.11's password:
Last login: Mon Jul 20 21:59:27 2015 from 172.123.25.222
[root@servidor1 ~]#
[root@servidor1 ~]# exit
logout
Connection to 172.123.25.11 closed.
[root@sinacceso ~]#
[root@sinacceso ~]#
[root@sinacceso ~]#
[root@sinacceso ~]# ssh -l jpueblo 172.123.25.12
jpueblo@172.123.25.12's password:
Last login: Mon Jul 20 21:59:53 2015 from 172.123.25.222
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@sinacceso ~]#
[root@sinacceso ~]# ssh -l ccalero 172.123.25.12
ccalero@172.123.25.12's password:
Last login: Mon Jul 20 22:00:43 2015 from 172.123.25.222
[ccalero@servidor2 ~]$
[ccalero@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@sinacceso ~]#
```

Figura 3.7. Intento de conexión de usuarios jpueblo y ccalero desde servidor no permitido

3.2.3. Sustitución de usuario root (su)

Para esta prueba debemos notar que es requisito conocer la clave del usuario **root** para poder tener éxito al ejecutar la sustitución con éxito.

En la primera imagen se muestra que el usuario **jpueblo** puede sustituir al usuario **root** en el **servidor1** sin novedades, pero en el **servidor2** no logra hacerlo.

```
[root@accesocentral1 ~]# ssh -l jpueblo 172.123.25.11
jpueblo@172.123.25.11's password:
Last login: Mon Jul 20 22:02:49 2015 from 172.123.25.111
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ su -
Password:
[root@servidor1 ~]#
[root@servidor1 ~]# exit
logout
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
[root@accesocentral1 ~]# ssh -l jpueblo 172.123.25.12
jpueblo@172.123.25.12's password:
Last login: Mon Jul 20 22:03:21 2015 from 172.123.25.111
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ su -
Password:
su: incorrect password
[jpueblo@servidor2 ~]$
[jpueblo@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.8. Intento de ejecución de su desde usuario jpueblo

En la segunda imagen se muestra que para el usuario **ccalero** en el **servidor1** no tiene sentido ejecutar la sustitución, dado que él ya tiene **uid** de **root**. En cambio en el **servidor2** el usuario **ccalero** si logra con éxito sustituir al usuario **root**.

```
[root@accesocentral1 ~]# ssh -l ccalero 172.123.25.11
ccalero@172.123.25.11's password:
Last login: Mon Jul 20 22:03:03 2015 from 172.123.25.111
[root@servidor1 ~]# id
uid=0(root) gid=0(root) groups=0(root)
[root@servidor1 ~]# pwd
/home/ccalero
[root@servidor1 ~]# exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
[root@accesocentral1 ~]# ssh -l ccalero 172.123.25.12
ccalero@172.123.25.12's password:
Last login: Mon Jul 20 22:03:39 2015 from 172.123.25.111
[ccalero@servidor2 ~]$
[ccalero@servidor2 ~]$ su -
Password:
[root@servidor2 ~]#
[root@servidor2 ~]# exit
logout
[ccalero@servidor2 ~]$
[ccalero@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.9. Intento de ejecución de su desde usuario ccalero

3.2.4. Ejecución de comandos restringidos usando sudo

En esta demostración se intentará usar el comando **service** para iniciar, detener y conocer el estado del servicio **httpd** en los servidores 1 y 2. Las pruebas se realizarán con los usuarios **jpueblo**, **ccalero** y **mleon**. Recordar que:

- el usuario **jpueblo** tanto en el **servidor1** como en el **servidor2** es un usuario sin privilegios especiales.

- el usuario **mleon** en el **servidor1** tiene uid de **root**, pero en el **servidor2** es un usuario al cual se le ha otorgado la

posibilidad de ejecutar ciertos comandos como **root**, excepto el comando **service**.

- el usuario **ccalero** en el **servidor1** tiene uid de **root**, pero en el **servidor2** es un usuario al cual se le ha otorgado la posibilidad de ejecutar ciertos comandos como **root**, incluido el comando **service**.

En las siguientes imágenes se muestran los intentos de ejecutar las acciones requeridas (conocer el estado, iniciar, detener) del servicio **httpd**, por cada usuario, primero en el **servidor1** e inmediatamente en el **servidor2**.

```
[root@accesocentral1 ~]# ssh -l jpueblo 172.123.25.12
jpueblo@172.123.25.12's password:
Last login: Tue Jul 21 18:26:41 2015 from 172.123.25.111
[jpueblo@servidor2 ~]$ sudo service httpd status

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for jpueblo:
jpueblo is not in the sudoers file.  This incident will be reported.
[jpueblo@servidor2 ~]$ sudo service httpd start
[sudo] password for jpueblo:
jpueblo is not in the sudoers file.  This incident will be reported.
[jpueblo@servidor2 ~]$ service httpd status
httpd is stopped
[jpueblo@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.10. Usuario jpueblo ejecutando comandos restringidos en servidor1

```
[root@accesocentral1 ~]# ssh -l jpueblo 172.123.25.11
jpueblo@172.123.25.11's password:
Last login: Mon Jul 20 22:13:48 2015 from 172.123.25.111
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ service httpd status
httpd is stopped
[jpueblo@servidor1 ~]$ service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for servidor1.msia3.local
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
(13)Permission denied: make_sock: could not bind to address [::]:80
(13)Permission denied: make_sock: could not bind to address 0.0.0.0:80
no listening sockets available, shutting down
Unable to open logs
[FAILED]
[jpueblo@servidor1 ~]$ service httpd status
httpd is stopped
[jpueblo@servidor1 ~]$
[jpueblo@servidor1 ~]$ exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
```

Figura 3.11. Usuario jpueblo ejecutando comandos restringidos en servidor2

```
[root@accesocentral1 ~]# ssh -l mleon 172.123.25.12
mleon@172.123.25.12's password:
Last login: Tue Jul 21 18:28:37 2015 from 172.123.25.111
[mleon@servidor2 ~]$ sudo service httpd status
[sudo] password for mleon:
Sorry, user mleon is not allowed to execute '/sbin/service httpd status' as root on servidor2.msia3.local.
[mleon@servidor2 ~]$ sudo service httpd start
[sudo] password for mleon:
Sorry, user mleon is not allowed to execute '/sbin/service httpd start' as root on servidor2.msia3.local.
[mleon@servidor2 ~]$ service httpd status
httpd is stopped
[mleon@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.12. Usuario mleon ejecutando comandos restringidos en servidor1

```
[root@accesocentral1 ~]# ssh -l mleon 172.123.25.11
mleon@172.123.25.11's password:
Last login: Tue Jul 21 18:22:52 2015 from 172.123.25.111
[root@servidor1 ~]#
[root@servidor1 ~]# service httpd status
httpd is stopped
[root@servidor1 ~]# service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for servidor1.msia3.local
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
[  OK  ]
[root@servidor1 ~]# service httpd status
httpd (pid  2350) is running...
[root@servidor1 ~]# service httpd stop
Stopping httpd: [  OK  ]
[root@servidor1 ~]# service httpd status
httpd is stopped
[root@servidor1 ~]#
[root@servidor1 ~]# exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
```

Figura 3.13. Usuario mleon ejecutando comandos restringidos en servidor2

```
[root@accesocentral1 ~]# ssh -l ccalero 172.123.25.12
ccalero@172.123.25.12's password:
Last login: Tue Jul 21 18:37:08 2015 from 172.123.25.111
[ccalero@servidor2 ~]$ sudo service httpd status
[sudo] password for ccalero:
httpd is stopped
[ccalero@servidor2 ~]$ sudo service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for servidor2.msia3.local
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
[  OK  ]
[ccalero@servidor2 ~]$ sudo service httpd status
httpd (pid  2495) is running...
[ccalero@servidor2 ~]$ sudo service httpd stop
Stopping httpd: [  OK  ]
[ccalero@servidor2 ~]$ sudo service httpd status
httpd is stopped
[ccalero@servidor2 ~]$ exit
logout
Connection to 172.123.25.12 closed.
[root@accesocentral1 ~]#
```

Figura 3.14. Usuario ccalero ejecutando comandos restringidos en servidor1

```
[root@accesocentral1 ~]# ssh -l ccalero 172.123.25.11
ccalero@172.123.25.11's password:
Last login: Tue Jul 21 18:09:36 2015 from 172.123.25.1
[root@servidor1 ~]#
[root@servidor1 ~]# service httpd status
httpd is stopped
[root@servidor1 ~]# service httpd start
Starting httpd: httpd: apr_sockaddr_info_get() failed for servidor1.msia3.local
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
[  OK  ]
[root@servidor1 ~]# service httpd status
httpd (pid  2279) is running...
[root@servidor1 ~]# service httpd stop
Stopping httpd: [  OK  ]
[root@servidor1 ~]# service httpd status
httpd is stopped
[root@servidor1 ~]#
[root@servidor1 ~]# exit
logout
Connection to 172.123.25.11 closed.
[root@accesocentral1 ~]#
```

Figura 3.15. Usuario ccalero ejecutando comandos restringidos en servidor2

CONCLUSIONES Y RECOMENDACIONES

Después del desarrollo anterior se concluye lo siguiente:

1. Es importante tener una Política de Seguridad definida desde la Gerencia.

2. Existen pequeñas modificaciones en archivos de configuración de los sistemas operativos que mejoran grandemente la seguridad de los usuarios.

3. La seguridad de la información depende del ambiente donde se va a aplicar. Por ello hay que definir acciones que se puedan ejecutar en nuestro ambiente.

4. Los usuarios rechazarán al inicio los cambios al aplicar una política, pero después se darán cuenta de los beneficios de usarla.

Para finalizar se recomienda:

1. Definir políticas de seguridad escribiéndolas, ayuda comentarlas, pero una vez escritas se pueden crear acciones concretas para cumplir las políticas definidas.

2. Antes de aplicar nuevas reglas crear un laboratorio, de ser posible virtualizado. No necesitará grandes recursos, sólo instalar lo estrictamente necesario para sus pruebas.

3. Concientizar a sus subalternos/compañeros acerca de usar siempre buenas prácticas respecto a la seguridad de los usuarios.

4. Ampliar este trabajo y/o las buenas prácticas que ya esté aplicando con herramientas de gestión adicionales: un sistemas de monitoreo de su infraestructura, LOGs centralizados, respaldos centralizados y de ser posible obtener las guías y buenas prácticas directamente de los fabricantes [6].

BIBLIOGRAFIA

[1] A. López Neira y J. Ruiz Spohr, «Portal de ISO 27001 en Español,» ISO27000.ES, [En línea]. Available: http://www.iso27000.es/iso27002_9.html. [Último acceso: 15 Julio 2015].

[2] T. Čapek y E. Deon Ballard, «RedHat Customer Portal,» Red Hat, 2015. [En línea]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Managing_Smart_Cards/PAM_Configuration_Files.html. [Último acceso: 1 Julio 2015].

[3] J. Turnbull, Hardening Linux, New York: Apress, 2005.

[4] M. Prpič, T. Čapek, S. Wadeley, Y. Ruseva, M. Svoboda y R. Krátký, «Red Hat Customer Portal,» Red Hat, 2014. [En línea]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/. [Último acceso: Julio 2015].

[5] TecMint.com, «TecMint,» TecMint.com, 3 Enero 2015. [En línea]. Available: http://www.tecmint.com/su-vs-sudo-and-how-to-configure-sudo-in-linux/. [Último acceso: Julio 2015].

[6] M. Prpič, T. Čapek, S. Wadeley, Y. Ruseva, M. Svoboda y R. Krátký, «Red Hat Customer Portal,» Red Hat, 2015. [En línea]. Available: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/. [Último acceso: Julio 2015].