CONTROL DE ACCESO
Práctica SL2
Informe de prácticas

1. Considerando los ficheros base que se emplean para almacenar las credenciales de usuario en los sistemas operativos de la familia UNIX, indique:

1. El nombre y ubicación de los mismos.

   En los sistemas de la familia UNIX se sigue el estándar de almacenar las credenciales de acceso de usuarios en el fichero **/etc/passwd**, que se complementa con **/etc/shadow** (donde se almacenan las claves) y **/etc/groups** (donde se almacenan los grupos de usuarios).

2. La relación existente entre los campos que se contienen en los mismos.

   El fichero **/etc/passwd** sigue una estructura de tabla diferenciando las columnas con el separador "**:**" y la fila mediante salto de línea. Los campos a los que se refiere cada una de las columnas son los siguientes:

   1. Nombre de usuario:
      Es el nombre del usuario en el sistema.

   2. Clave de acceso:
      La clave de acceso de dicho usuario. Si en este campo aparece una **x** esto indica que la clave se encuentra en el fichero **/etc/shadow**. Posteriormente hablaremos de dicho fichero.

   3. Identificador de usuario:
      Se corresponde con el número de identificador del usuario. El valor 0 está reservado para root, del 1 al 99 para usuarios predefinidos del sistema y del 100 en adelante para las cuentas de usuario

   4. Identificador de grupo:
      Número que representa el identificador del grupo primario del usuario. Este número hace referencia al fichero **/etc/groups**.

   5. Descripción de usuario:
      Campo de texto destinado a añadir más información sobre el usuario como nombre completo o descripción.

   6. Directorio Home:
      Indica la ruta absoluta al directorio home de dicho usuario.

   7. Shell de comandos:
      Indica la ruta absoluta del Shell de comandos que por defecto utilizará el usuario.

El fichero **/etc/shadow** es el encargado de almacenar las claves de usuario, el cual sigue una estructura de tabla al igual que el anterior. En este caso los campos son los siguientes:

1. Nombre de usuario:
   Es el nombre del usuario en el sistema.

2. Clave de Acceso:
   Clave de acceso del usuario, encriptada. El algoritmo por defecto es DES pero se pueden utilizar otros como MD5 (en este caso se anotaría 1 al comienzo). En el caso de que este campo sea "!" o "*" quiere decir que se utilizan otras vías para la autenticación. En el caso de la exclamación quiere decir que está protegida.

3. Fecha del último cambio:
   Fecha de la última modificación de la clave de usuario.

4. Mínimo:
   Mínimo número de días necesario para poder volver a cambiar la clave.

5. Máximo:
   Máximo número de días necesario para poder volver a cambiar la clave.

6. Número de días para cambiar la clave:
   Número de días desde que al usuario se le avisa para cambiar la clave antes de que esta deje de ser válida

7. Número de días antes para la eliminación de la cuenta
   Número de días tras los cuales se eliminará la cuenta en el caso de no haber cambiado la clave en el periodo anterior.

8. Fecha de eliminación de la cuenta:
   Fecha tras la cual la cuenta será eliminada.

El fichero **/etc/group** es el encargado de almacenar el listado grupos de usuario, el cual sigue una estructura de tabla al igual que los anteriores. En este caso los campos son los siguientes:

1. Nombre de grupo:
   Es el nombre del grupo en el sistema.

2. Clave de Acceso:
   Es la clave de acceso al grupo. Puede estar alojada en **/etc/shadow** al igual que las claves de usuario.

3. Identificador de grupo:
   Número de identificación del grupo en el sistema.

4. Lista de usuarios:
   Listado de los nombres de usuario de los usuarios que pertenecen a dicho grupo separados por comas.

3. Cuáles serían los modos de acceso esperados para estos ficheros y qué tipo de amenazas se podrían esperar cuando los modos no sean los adecuados.

   El fichero **/etc/passwd** tiene como propietario root, el cual tiene permisos de lectura y escritura, y está en el grupo root. Lo mismo sucede con **etc/group**, En el caso de /**etc/shadow** este pertenece al grupo shadow. En cuanto a los modos de acceso para el resto de usuarios tanto passwd como group permiten lectura, sin embargo, shadow no lo permite.

   Los tipos de amenazas esperadas en este tipo de ficheros podrían ser: la modificación/eliminación de claves de usuario para poder acceder a dichos usuarios/grupos sin restricciones, añadir a usuarios a determinados grupos, o la perturbación de otros campos. El método para conseguir llevar a cabo este tipo de ataques se podría llevar a cabo obteniendo privilegios de usuario root por otras vías, y por tanto, pudiendo modificar estos ficheros.

2. En relación con los modos de acceso que se emplean para controlar el acceso a los ficheros en un entorno operativo de la familia UNIX, indique:

1. Los niveles en que se agrupan, los modos de acceso que se soportan, la forma de indicar esos modos en los listados de ficheros y en las órdenes que permiten modificarlos.

   El sistema Unix clasifica los niveles de acceso en tres grupos: el propietario, el grupo y el resto de usuarios. Los modos de acceso que se soportan también son tres: permiso de lectura, de escritura y de ejecución.

   Estos modos se suelen representar de dos formas, la primera sería la extendida, que consiste de 9 caracteres, de los cuales los 3 primeros corresponden a los permisos del propietario, los 3 siguientes a los del grupo y los restantes al resto de usuarios. Se asignan 3 caracteres a cada nivel de acceso para poder representar lectura, escritura y ejecución. Estos se designan con rwx. Un ejemplo de esta representación es:

   rwxr-xr--

   Existe otra representación más sucinta, que consiste en representar estos 9 caracteres como un número en base 8, por tanto. el ejemplo anterior se podría representar como:

   761

2. Describa brevemente la misión de las órdenes `chown`, `chmod`, `chattr`.

   Las órdenes existentes para modificar estos ficheros son los siguientes:
   - chown: permite modificar el propietario de un fichero.
   - chgrp: permite modificar el grupo de un fichero.
   - chmod: permite modificar todos los permisos de un fichero.
   - chattr: permite modificar los atributos de un fichero. Estos pueden ser hacerlo inmutable, no eliminable, etc.

3. Describa con un ejemplo la utilidad de la orden `umask` e indique cuál sería el lugar indicado para invocarla en el contexto de una sesión de usuario.

   La orden umask sirve para decidir la política de modos de acceso de un nuevo fichero creado por un proceso. Un ejemplo de uso sería el siguiente:

   Por defecto se suele asignar 0022, que podemos visualizar ejecutando umask sin argumentos. Esto quiere decir que los nuevos ficheros podrán ser leídos por cualquiera, pero solo modificados por el propietario. Si ahora ejecutamos la orden umask 0077 entonces habremos eliminado los permisos de lectura para el grupo y otros para los nuevos ficheros que creemos. Un ejemplo de esto se recoge en la siguiente captura de pantalla:

```
mallet@mallet:~$ umask
0022
mallet@mallet:~$ touch new-file
mallet@mallet:~$ ls -l new-file
-rw-r--r-- 1 mallet mallet 0 2016-10-06 18:52 new-file
mallet@mallet:~$ umask 0077
mallet@mallet:~$ touch new-file-2
mallet@mallet:~$ ls -l new-file-2
-rw------- 1 mallet mallet 0 2016-10-06 18:52 new-file-2
mallet@mallet:~$
```

   En el contexto de una sesión de usuario se puede ejecutar en cualquier parte ya que esta orden no afecta a directorios, sino a procesos. Por tanto, una manera para que estos cambios se hicieran efectivos y no desaparecieran al cerrar el Shell tendríamos que añadir dicha orden en el fichero **~/.profile** para que este se ejecutara siempre.

3. Indique cómo usaría la orden `chroot` y la configuración de una cuenta de usuario en UNIX de forma que se pueda controlar que sólo pueda acceder a los ficheros de su carpeta de usuario, tanto para ver o modificar como para ejecutar.

   Para realizar estas operaciones tendría que ejecutar la orden chroot indicando el directorio que se utilizaría como raíz, por lo general **/home/username**.

   Además, para que esto pudiera funcionar necesitaría que en este directorio se encontrase un bash y un ls.

```
alice@alice:~$ mkdir /home/proof/bin /mkdir/proof/lib
```

   Para ello una solución sería copiarles de /bin/bash y /bin/ls en el directorio **/home/username/bin**

```
alice@alice:~$ cp /bin/bash /bin/ls /home/proof/bin
```

   Para que estos ejecutables funcionen apropiadamente necesitaríamos apoyarnos en algunas librerías que deberíamos almacenar en **/home/username/lib**. Para conocer las dependencias necesarias para estas ordenes podemos usar el comando ldd, que nos muestra el listado de librerías utilizadas:

```
alice@alice:~$ ldd /bin/bash
        linux-gate.so.1 =>  (0x00ea2000)
        libncurses.so.5 => /lib/libncurses.so.5 (0x00ce7000)
        libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0x006a1000)
        libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0x00110000)
        /lib/ld-linux.so.2 (0x007e7000)
alice@alice:~$ ldd /bin/ls
        linux-gate.so.1 =>  (0x004cf000)
        librt.so.1 => /lib/tls/i686/cmov/librt.so.1 (0x00e4d000)
        libselinux.so.1 => /lib/libselinux.so.1 (0x0092f000)
        libacl.so.1 => /lib/libacl.so.1 (0x00b3d000)
        libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0x00110000)
        libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0 (0x0026a000)
        /lib/ld-linux.so.2 (0x00822000)
        libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0x00ded000)
        libattr.so.1 => /lib/libattr.so.1 (0x00f32000)
```

Una vez hecho esto podríamos ejecutar la orden con permisos root y entraríamos en un bash que tan solo permitiría el acceso a la carpeta del usuario

```
alice@alice:~$ sudo chroot /home/proof
```

4. Describa cómo usaría el cliente `telnet` para determinar qué conjunto de servicios tiene abiertos `alice` desde `mallet`.

Para poder realizar una conexión telnet desde mallet hacia alice lo primero que haría sería comprobar si su dirección ip está en /etc/hosts. Una vez hecho esto ejecuté la siguiente orden:

```
mallet@mallet:~$ telnet alice
```

Seguidamente introduje los datos de usuario de alice para por último ejecutar el comando service, con el cual se obtiene un listado completo de todos los servicios que hay en el sistema y su estado actual:

```
alice@alice:~$ service --status-all
 [ ? ]  acpi-support
 [ ? ]  acpid
 [ ? ]  alsa-mixer-save
 [ ? ]  anacron
 [ + ]  apache2
 [ - ]  apparmor
 [ ? ]  apport
 [ ? ]  atd
 [ ? ]  avahi-daemon
 [ ? ]  binfmt-support
 [ - ]  bluetooth
 [ - ]  bootlogd
 [ - ]  brltty
 [ ? ]  console-setup
 [ ? ]  cron
 [ - ]  cups
 [ ? ]  dbus
 [ ? ]  dmesg
 [ ? ]  dns-clean
 [ - ]  exim4
 [ ? ]  failsafe-x
 [ - ]  fancontrol
 [ ? ]  gdm
 [ - ]  grub-common
 [ ? ]  gssd
```

5. Documente paso a paso cómo discurre el proceso de conexión a `alice` desde `mallet` vía SSH.

Una vez generadas las claves RSA para permitir la autenticación sin necesidad de tener que escribir las contraseñas utilizando el fichero **.ssh/authorized_keys** ya está todo preparado para conocer paso a paso como es el proceso de conexión entre alice y mallet lo primero que se debe hacer es activar el modo debug del servidor ssh. En este caso dado que la conexión será de mallet a alice esta operación tendrá que realizarse en alice:

```
alice@alice:~$ sudo stop ssh
ssh stop/waiting
alice@alice:~$ sudo /usr/sbin/sshd -d 2>&1 | less
```

Una vez hecho esto ya tenemos activado el modo depuración (en este modo solo se puede realizar una conexión por cada vez que se ejecuta ssh). Por tanto, el siguiente paso es establecer la conexión de mallet (enviando su clave para verificar su identidad):

```
mallet@mallet:~$ ssh -i mallet-key alice@alice
Linux alice 2.6.32-28-generic #55-Ubuntu SMP Mon Jan 10 21:21:01 UTC 2011 i686 GNU/Linux
Ubuntu 10.04.2 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

Last login: Sat Oct  8 14:56:21 2016 from mallet
Environment:
  LANG=en_US.UTF-8
  USER=alice
  LOGNAME=alice
  HOME=/home/alice
  PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
  MAIL=/var/mail/alice
  SHELL=/bin/bash
  SSH_CLIENT=192.168.1.3 50402 22
  SSH_CONNECTION=192.168.1.3 50402 192.168.1.2 22
  SSH_TTY=/dev/pts/1
  TERM=xterm
  XDG_SESSION_COOKIE=ce821a6f031e5c9e594612674c2dba12-1475931401.260509-1238586257
alice@alice:~$
```

Sabemos que el modo depuración ha sido activado correctamente dado que ya nos aparecen campos que antes no se mostraban como el "environment" del host alice. Pero donde se puede visualizar el conjunto de pasos que han transcurrido al establecer la conexión es en el Shell de alice:

```
Connection from 192.168.1.3 port 44918
debug1: Client protocol version 2.0; client software version OpenSSH_5.3p1 Debian-3ubuntu5
debug1: match: OpenSSH_5.3p1 Debian-3ubuntu5 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu5
debug1: permanently_set_uid: 115/65534
debug1: list_hostkey_types: ssh-rsa,ssh-dss
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: client->server aes128-ctr hmac-md5 none
debug1: kex: server->client aes128-ctr hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST received
debug1: SSH2_MSG_KEX_DH_GEX_GROUP sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_INIT
debug1: SSH2_MSG_KEX_DH_GEX_REPLY sent
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: KEX done
debug1: userauth-request for user alice service ssh-connection method none
debug1: attempt 0 failures 0
```

Lo que sucede durante el transcurso del establecimiento de conexión es que el host cliente indica el tipo de máquina que es, así como el protocolo que desea utilizar durante la conexión. El establecimiento de conexión se divide en dos partes, primero se configura la clave de encriptado de sesión que será la que se utilizará durante todo el transcurso de la misma para que la transferencia de datos sea segura. Normalmente se utiliza el algoritmo Diffie-Hellman para generar la clave y AES como algoritmo de cifrado simétrico.

Una vez completado este proceso se procede a verificar el acceso del cliente al sistema. En este punto es donde se comprueba la contraseña de acceso al sistema en el caso de no utilizar pares de claves RSA como es nuestro caso, por lo tanto, se verifica con la clave pública entregada por mallet anteriormente alojada en **.ssh/authorized_keys** que es él enviando un mensaje cifrado con la misma y esperando que lo devuelva descifrado (En realidad envía un resumen generado con MD5). Una vez finalizado este proceso la conexión queda establecida y mallet puede ejecutar ordenes en un Shell de alice.

6. ¿Qué ventajas reporta el uso de `ssh-agent` combinado con `ssh-add` para acceder a un anfitrión remoto usando el mecanismo basado en clave pública-privada?

La principal ventaja que aporta el uso de dichos comandos junto con el mecanismo de autenticación es el incremento de seguridad debido al uso de claves criptográficas seguras mucho más inmunes al descifrado por fuerza bruta (ya que son generadas de forma aleatoria, algo que no suele suceder con las contraseñas de usuario) junto con la facilidad de uso una vez configuradas.

La idea consiste en que cada host (o usuario) tenga una clave privada que le identifica. Además de esto también posee una clave publica que utilizará para compartir entre el resto de usuarios. Estas claves normalmente se generan con el algoritmo de clave publica-privada RSA. Para ello existe una orden que simplifica el proceso: ssh-keygen

El mecanismo de claves funciona de la siguiente manera: Un usuario que desee permitir el acceso de este en su sistema tendrá que almacenar su clave pública en el fichero **.ssh/authorized_key**s y una vez hecho esto, este usuario podrá acceder con su clave privada en vez de con la contraseña del usuario.

La utilidad ssh-add sirve para enlazar automáticamente la clave a todas las peticiones ssh que se hagan desde el host para no tener que añadirla como opción a cada conexión.

Una vez completados dichos pasos, se podrá acceder al host destino sin escribir la clave y de forma segura. La única vulnerabilidad que puede ocurrir es la sustracción de la clave privada de otro usuario, por lo que se podría acceder a otros hosts suplantando su identidad.

7. Estudie documentación disponible en Internet y elabore una recomendación de configuración segura de un servidor SSH y describa las directivas de configuración esenciales y la finalidad que cumplen.

Algunas de las principales recomendaciones para aumentar la seguridad de un servidor SSH son las siguientes:

- Usar contraseñas largas y seguras, preferiblemente generadas de forma aleatoria para así dificultar los ataques basados en fuerza bruta que se apoyan en diccionarios de

claves.

- Deshabilitar el acceso root mediante ssh para tratar de limitar cambios de configuración del sistema de manera remota. Para ello se debe modificar el fichero **/etc/ssh/sshd_config** y añadir la siguiente línea: "`PermitRootLogin no`"
- Limitar el conjunto de usuarios que pueden conectarse a través de SSH. Esta práctica es una especie de whitelist de usuario para el servidor ssh. Para llevarlo a cabo se debe modificar el fichero **/etc/ssh/sshd_config** añadiendo la línea: "`AllowUsers usuario1 usuario2`"
- Restringir el uso del protocolo 1 de ssh ya que se han encontrado vulnerabilidades que permiten obtener el tráfico de datos a partir de un ataque man-in-the-middle. Por lo que deberemos forzar el uso del protocolo 2. Para llevarlo a cabo se debe modificar el fichero **/etc/ssh/sshd_config** añadiendo la línea: "`Protocol 2`"
- Usar un puerto diferente del predefinido para tratar de protegerse ante estrategias de ataque como escaneos de puertos para conocer si existe un servidor ssh corriendo en el sistema objetivo.
- Filtrar las conexiones ssh con un firewall pudiendo restringir las direcciones IP externas con la posibilidad de poder conectarse mediante ssh.
- Utilizar claves pública-privada para el acceso ssh como se ha explicado en ejercicios anteriores. Además, se puede activar la opción para prohibir el acceso con contraseña clásica. Esto se puede realizar añadiendo en el fichero **/etc/ssh/sshd_config** la línea "`PasswordAuthentication No`"

NOTAS SOBRE LA EVALUACIÓN:

i. Se valorará la precisión y corrección de las respuestas.
ii. Podrá emplear cuantas figuras desee para ilustrar su respuesta, siempre y cuando aporten información valiosa o sean realmente ilustrativas.
iii. En las cuestiones 5, 6 y 7 se valorará especialmente que no se limite a trasladar la información que haya encontrado en fuentes externas sino que haga un planteamiento claro y crítico de su propuesta. Estas cuestiones tienen un valor doble que las anteriores.
iv. La calificación final será sobre 10 = P1+P2+P3+P4+2*(P5+P6+P7)