Capítulo 10

Apéndices

Descripción de los módulos PAM

Certificados digitales

Script Qmail.sh

Bibliografía

Capítulo 10 Apéndices

En esta parte se han añadido aquellos contenidos que por su densidad o porque rompían el hilo narrativo no se pudieron escribir en su momento.

No es necesario llegar a asimilar los apéndices para la comprensión del proyecto en sí, pero si queremos profundizar forman un material de apoyo elemental.

Además hemos incluido un script que consideramos básico. Se trata de la instalación y configuración de qmail. En la memoria del proyecto hemos indicado como instalar cada paquete. El único que faltaba era qmail. Incluyéndolo como parte del apéndice tratamos de seguir el mismo criterio. Evidentemente se han usado más scripts pero que no incluimos como parte de la memoria. Se encuentran en el cd adjunto ampliamente comentados. Son importantes pero no nos interesan el "cómo" lo hacen, sino sólo los resultados. En el caso de qmail.sh si nos interesa, de ahí su inclusión.

10.1 Descripción del uso de cada componente de los módulos PAM

pam_unix

Es el módulo de autenticación estándar de Unix. Usa llamadas estándar para extraer y establecer tanto información de contabilidad como de autenticación. Normalmente obtiene esta información del fichero /etc/passwd o del /etc/shadow si tenemos activo shadow.

Componente Account

Argumentos reconocidos: debug ; audit

Descripción

El argumento debug hace que se recoja más información de contabilidad. El argumento audit hace que se recoja aún más.

Basándose en los siguientes elementos de shadow: expire; last_change; max_change; min_change; warn_change, este módulo ejecuta la tarea de establecer el estatus de la cuenta de usuario y de su password.

Componente Auth

Argumentos reconocidos: debug; audit; use_first_pass ; try_first_pass; nullok; nodelay.

Descripción

Los argumentos debug y audit tienen las mismas funciones que en la componente anterior.

La acción por defecto de este módulo es no permitir al usuario el acceso al servicio si su clave oficial está en blanco. El argumento nullok anula este comportamiento.

Cuando aparece el argumento try_first_pass, antes de pedir el password al usuario, el módulo intenta autenticar mediante la clave suministrada anteriormente en algún módulo apilado. Si aparece use_first_pass se fuerza al módulo a que use este password y por tanto no lo pedirá de nuevo al usuario. Así que si no tenemos clave o ésta no es válida, se le denegará el acceso al usuario.

El argumento nodelay se usa para deshabilitar un retraso (de más o menos un segundo) que existe cuando surge un fallo en la autenticación.

Componente Passwd

Argumentos reconocidos: debug, audit; nullok; not_set_pass; use_authtok; try_first_pass; use_first_pass; md5; bigcrypt; shadow; nis; remember.

Descripción

Esta parte del módulo pam_unix se encarga de la tarea de actualizar el password de usuario.

En el caso de bases de datos convencionales de Unix (que almacenan el password encriptado), el argumento md5 se usa para realizar la encriptación mediante la función MD5, opuesta a la llamada convencional crypt(3). Una alternativa a esto es usar bigcrypt, una extensión del algoritmo crypt.

El argumento nullok se usa para permitir el cambio de un password desde uno que está en blanco. Sin este argumento, los passwords en blanco son tratados como bloqueos de cuenta.

El argumento use_first_pass se usa para bloquear la elección de antiguas y nuevas contraseñas según establezca el módulo previo apilado de tipo passwd. El argumento try_first_pass se usa para evitar que el usuario tenga que reintroducir el password cuando pam_unix sigue a un módulo que posiblemente comparta el antiguo password. De este modo si este password antiguo no es válido se pide que se introduzca el correcto.

El argumento use_authtok se usa para forzar a este módulo a establecer el nuevo password al que provee el anterior módulo apilado de tipo passwd.

Con el argumento nis, pam_unix intentará usar NIS RPC para establecer los nuevos passwords.

El argumento remember toma un valor, que es el número más reciente de passwords que debe recordar para cada usuario. Se guardan en el fichero

/etc/security/opasswd y sirven para forzar el cambio de contraseñas del usuario y así no repita las mismas frecuentemente.

Componente Session

No se reconoce ningún argumento en este componente. Su cometido se limita a guardar logs del nombre de usuario y del tipo de servicio.

pam_cracklib

Este módulo se apila para proveer un chequeo de la robustez de los passwords establecidos. Trabaja de la siguiente forma:

Primero llama a la rutina Cracklib para comprobar la fortaleza del password. Si pasa este primer test se le somete a otro adicional. Estos tests son:

- Palíndromo.
  ¿es el nuevo password un palíndromo del anterior?
- Cambio de caso.
  ¿Es el nuevo password el mismo que el anterior salvo un pequeño cambio de caso?
- Similar.
  ¿Es el nuevo password demasiado similar al anterior?
- Simple.

¿Es el nuevo password demasiado pequeño? Esto es controlado por cinco argumentos: minlen,dcredit, ucredit,lcredit y ocredit.

- Rotado.
  ¿Es el nuevo password una versión rotada del anterior?
- Ya usado.

¿Fue usado en el pasado? Como ya hemos visto, los passwords usados pueden encontrarse en /etc/security/opasswd.

El módulo sin argumentos trabaja bien para una encriptación estándar. Si usamos encriptación md5, las claves pueden ser mayores de 8 caracteres, y las condiciones de elección de un nuevo password se endurecen bastante.

Componente Passwd

Argumentos reconocidos: Debug; type=XXX; retry=N; difok=N; minlen=N; dcredit=N; ucredit=N; lcredit=N; ocredit=N; use_authtok.

Descripción

La acción de este módulo es pedir al usuario un password y chequear su fortaleza contra un ataque de diccionario y una serie de reglas de identificación de malas elecciones.

Por defecto se pide un password, se chequea su fortaleza y entonces, si se considera suficientemente seguro, se pide por segunda vez (para verificar que fue correctamente escrito). Este comportamiento por defecto se puede cambiar usando los argumentos que vimos al principio.

- debug

Esta opción hace que el módulo escriba información a syslog(3).

- type=XXX

  Por defecto el módulo usa la siguiente frase cuando pide una contraseña: “New UNIX password: “ y “Retype UNIX password: ”. Si usamos esta opción podemos reemplazar la palabra UNIX por XXX.

- Retry=N

N es el número de veces que el módulo pedirá un nuevo password para comprobar su fortaleza. Por defecto es 1.

- difok=N

Este argumento cambia el número de caracteres que no deben estar presentes en el nuevo password que sean del antiguo. Por defecto es 10, pero con tal de que la mitad de los caracteres sean diferentes, el password se acepta.

- minlen=N

Es el tamaño mínimo aceptable para el nuevo password (más uno, si los créditos no están deshabilitados, que es el comportamiento por defecto). Además del número de caracteres en el nuevo password, tenemos cierto crédito (que será una cantidad de un carácter en longitud), que se da por cada diferente tipo de carácter (mayúsculas, minúsculas, dígitos y otros).

Por defecto este parámetro es 9, que está bastante bien para el antiguo estilo de las contraseñas en Unix, pero que no explota todo lo que da de sí el sistema md5. El concepto crédito quedará claro en los siguientes puntos.

- dcredit=N (con N>=0)

Este es el máximo crédito por tener dígitos en el nuevo password. Si tiene menos de N dígitos cada uno de éstos contará como un carácter más que deberá tener el valor de minlen. Por defecto N=1, que sería el mínimo número de dígitos que debe tener un nuevo password.

- ucredit=N (con N>=0)

Es el máximo crédito por tener mayúsculas en el password nuevo. Si tiene menos o N mayúsculas cada una contará como un carácter más en el valor minlen. Por defecto N=1, que es el valor recomendado para un minlen menor que 10.

- lcredit=N (con N>=0)

Es el máximo crédito por tener minúsculas en el nuevo password. Si tiene menos o N minúsculas cada letra contará como un carácter más en el valor de minlen. Por defecto N=1.

- ocredit=N (con N>=0)

Es el máximo crédito por tener otros caracteres en el nuevo password. Si tiene menos o N caracteres de este tipo cada carácter contará como uno más en el valor de minlen. Por defecto N=1.

- use_authtok

Este argumento se usa para forzar al módulo a no pedir al usuario de nuevo el password, sino a usar el que provee el módulo apilado previamente.

pam_deny

Este módulo se utiliza para denegar el acceso. Devuelve un código de error cuando la autenticación falla.

Componente Account

Este componente no hace otra cosa que devolver un fallo. El tipo de fallo es PAM_ACCT_EXPIRED.

Componente Auth

Hace lo mismo que Account con la diferencia del tipo de fallo. En este caso es PAM_AUTH_ERROR.

Componente Passwd

Esta componente deniega al usuario la oportunidad de cambiar su password. Siempre responde con PAM_AUTHTOK_ERR cuando se invoca.

Componente Session

Esta parte previene a la aplicación de empezar una sesión.

pam_env

Este módulo permite establecer variables de entorno.

Argumentos reconocidos: debug; conffile=archivo; envfile=archivo; readenv=0/1.

Componente Auth

Permite establecer o borrar variables de entorno arbitrariamente, usando cadenas fijas, valores previos de las variables o PAM_ITEMs.

Todo se controla mediante el fichero de configuración (por defecto / etc/security/pam_env.conf, pero que puede ser sustituido por el argumento conffile). Cada línea de este fichero comienza con el nombre de la variable, entonces hay dos posibilidades para cada variable, DEFAULT y OVERRIDE. DEFAULT permite al administrador establecer el valor por defecto de la variable. La opción OVERRIDE indica a pam_env

que debería introducir el valor (sobreescribiendo el valor por defecto), si hay alguno que usar.

VARIABLE [DEFAULT=[value]] [OVERRIDE=[value]]

El comportamiento de este módulo se puede modificar con:

- debug. Escribe más información a syslog.
- envfile=archivo. Por defecto se usa el archivo /etc/environment para cargar los pares KEY=VAL directamente en el entorno. Con esta opción podemos especificar otro archivo.
- Readenv=0/1. Habilitamos (1) o deshabilitamos (0) la lectura del archivo especificado en envfile. Por defecto está a habilitada (1).

pam_filter

Componente Auth, Account, Passwd y Session

Argumentos reconocidos: debug; new_term.

Cada componente del módulo tiene el potencial para invocar al filtro deseado. El filtro se ejecuta con el privilegio de la aplicación que llama y no con el del usuario. Por esta razón no se puede matar sino es cerrando la sesión.

El comportamiento del módulo es sensiblemente alterado mediante los siguientes argumentos:

- debug. Esta opción incrementa la cantidad de información que se le pasa a syslog para escribir los logs.
- new_term. Por defecto, el filtro establece el objeto PAM_TTY para indicar el terminal que el usuario está usando para conectarse a la aplicación. Este argumento indica que el filtro debería establecer PAM_TTY al pseudo-terminal filtrado.
- non_term. Este argumento indica que no se intente establecer el objeto PAM_TTY.

pam_group

Este módulo provee una configuración de grupo basada en el nombre de usuario y en el terminal desde el que se está pidiendo el servicio en cuestión. Además toma nota de la hora del día en la que se realiza la petición.

Componente Auth

Este módulo no autentica usuarios, sino que otorga membrecía a grupos al usuario. Tales membrecías se basan en el servicio en el que se aplica. Las membrecías de grupo se listan en el fichero /etc/security/group.conf.

La sintaxis de este fichero es la siguiente:

services ; ttys ; users ; times ; groups

Por último decir que el módulo pam_group funciona en paralelo con el fichero /etc/group. Si al usuario se le concede pertenecer a algún grupo, según el comportamiento del módulo, se le concede también las entradas equivalentes en dicho fichero /etc/group.

pam_lastlog

Este módulo se encarga de mantener el archivo /var/log/lastlog. Añade una entrada a dicho archivo cuando se abre una sesión, de forma que puede proveer información sobre la última sesión que abrió un usuario en particular.

Componente Session

Argumentos reconocidos: debug; nodate; noterm; silent; never.

El mensaje que añade a /var/log/lastlog es el típico de “Last login on ...” cuando el usuario accede al sistema.

El comportamiento de este módulo puede modificarse gracias a sus argumentos:

- debug. Escribe más información a syslog.
- nodate. No da la fecha del último login en el sistema.
- noterm. No indica el nombre del terminal en donde fue el último login.
- nohost. No indica el último equipo desde el que se hizo el último login.
- silent. No informa al usuario de ningún dato sobre el último login. Sólo actualiza el fichero /var/log/lastlog.
- never. Si el fichero /var/log/lastlog no contiene ninguna entrada para el usuario, entonces el módulo indica que nunca antes había accedido a esta máquina y le ofrece un mensaje de bienvenida.

pam_limits

Este módulo permite establecer límites en los recursos del sistema que puede obtener un usuario en su sesión. Estos límites se fijan en el fichero de configuración /etc/security/limits.conf.

Componente Session.

Argumentos reconocidos: debug; conf=archivo; change_uid.

A través de los contenidos del fichero de configuración podemos establecer límites en las sesiones de los usuarios. Aquellos con uid=0 no les afecta ninguna restricción.

El comportamiento del módulo puede verse afectado mediante los siguientes argumentos.

- debug. Escribe logs a syslog para poder depurar el comportamiento.
- conf=archivo. Indica un archivo alternativo donde especificar la configuración.
- change_uid. Cambia la uid real del usuario para el que los límites están establecidos. No se suele usar.

La sintaxis del fichero /etc/security/limits.conf es la siguiente:

<domain> <type> <item> <value>

Cada campo puede contener diversos objetos.

<domain> puede ser:

- un nombre de usuario
- un nombre de grupo, con la sintaxis siguiente: @group
- un carácter de escape *, que correspondería a la entrada por defecto.
- Un carácter de escape %, que sería sólo para el límite maxlogin. Puede ser usado con la sintaxis %gruop

<type> puede tener tres valores:

- hard. Los límites hard son establecidos por el superusuario y respaldados por el kernel. El usuario no puede pedir más de los recursos del sistema que aparecen en estos límites.
- soft. Los límites soft son aquellos que el usuario puede incrementar o decrementar dentro de un rango y siempre por debajo de los límites hard preexistentes. Los límites especificados con este valor pueden ser tomados como los límites por defecto, en un uso normal del sistema.
- -. Mediante un guión (-) indicamos ambos límites simultáneamente.

<item> puede ser uno de los siguientes:

- core. Límite del tamaño del fichero de núcleo (KB).
- data. Máximo tamaño de los datos (KB).
- fsize. Tamaño máximo de fichero (KB).
- memlock. Máximo tamaño de direcciones de memoria para bloquear KB.
- nofile. Número máximo de ficheros abiertos.
- stack. Máximo tamaño de pila (KB).
- cpu. Máximo tiempo de cpu (min).
- nproc. Máximo número de procesos.
- as. Límite del espacio de direcciones.
- maxlogins. Máximo número de logins para este usuario.
- maxsyslogins. Máximo número de logins en el sistema.
- priority. Prioridad para ejecutar procesos de usuario.
- locks. Máximo número de ficheros bloqueados.

pam_nologin

Este módulo provee autenticación sin introducir el login. Si el archivo /etc/nologin existe, sólo el root puede acceder; los otros usuarios son rechazados mediante un código de error.

Componente Auth

Argumentos reconocidos: successok; file=archivo.

El superusuario puede anular el fichero por defecto /etc/nologin y especificar otro mediate el argumento file. Si el fichero no existe, el módulo por defecto devuelve PAM_IGNORE, pero si tenemos successok nos devolverá PAM_SUCCESS.

pam_permit

Este módulo sólo sirve para permitir accesos, no hace nada más. De hecho se conoce como el módulo promiscuo. Debe ser usado con mucha precaución.

Componentes Account, Auth, Passwd y Session

No importa que tipo de grupo sea, la acción de este módulo es simplemente devolver el código de operación exitosa, PAM_SUCCESS.

pam_rootok

Este módulo se utiliza en situaciones en las que el superusuario quiere obtener acceso a un servicio sin tener que introducir el password.

Componente Auth.

Argumento reconocido: debug.

Esta componente autentica al usuario si su uid es 0. Debemos tener en cuenta que es la uid real la que se chequea y no la efectiva.

No se debería usar este módulo para programas que ejecute el superusuario o que se inician con el sistema por motivos de seguridad.

pam_securetty

Provee del chequeo del estándar securetty de Unix, que causa que la autenticación para el root falle a menos que PAM_TTY esté establecida a una cadena listada en el fichero /etc/securetty. Para el resto de usuarios devuelve éxito.

Componente Auth

Para un uso correcto debería ser listado como un método required antes de los métodos de autenticación sufficient.

pam_pwdb

Este módulo podría sustituir a cualquiera de la serie pam_unix_... con la diferencia de que usa la interfaz genérica de la biblioteca de base de datos de passwords libpwdb.

Componente Account

Argumentos reconocidos: debug

Descripción

El argumento debug hace que se recoja más información de contabilidad.

Componente Auth

Argumentos reconocidos: debug; use_first_pass; try_first_pass; nullok; nodelay; likeauth.

Descripción

Estos argumentos producen los mismos efectos que en la componente Auth del módulo pam_unix, así que no lo vamos a comentar.

Componentes Password y Session

En estas componentes ocurren lo mismo que en la anterior. Los argumentos son los mismos que en pam_unix y su significado también, por lo que tampoco se comentarán.

pam_tally

Este módulo mantiene un contador de intentos de accesos. Puede resetear este contador en caso de éxito y también denegar el acceso si tenemos demasiados intentos fallidos.

Componente Auth

Argumentos reconocidos: onerr=(succeed | fail); file=

Descripción

Esta componente lo único que realiza es el incremento de intentos en el contador. El argumento onerr permite establecer el comportamiento del módulo en el caso de que algo extraño ocurra. Podemos decidir que lo deje pasar (succeed) o que de error esa componente (fail). Además podemos especificar el fichero donde se encuentre los contadores. Si no se indica nada, por defecto es /var/log/faillog.

Componente Account

Argumentos reconocidos: onerr=(succeed | fail); file= ; deny=n; reset.

Descripción

La componente Account puede denegar el acceso y/o resetear el contador de intentos. Además comprueba que el contador es un fichero de texto plano y que no puede escribir en él todo el mundo.

Los argumentos onerr y file tienen el mismo significado que en la componente Auth.

El argumento deny permite a tally denegar el acceso al usuario que sobrepasa los n intentos. La opción reset le indica al módulo que vuelva el contador a 0 cuando tengamos una entrada correcta.

pam_time

Este módulo basa su funcionamiento en el control del tiempo para permitir accesos a los servicios gestionados.

Componente Account

Descripción

A veces tenemos que restringir el acceso a determinados servicios basándonos en el horario. Podemos denegar el acceso a usuarios teniendo en cuenta sus nombres, la hora del día, el día de la semana, el servicio que están demandando y el terminal desde el que están haciendo dicha petición.

El módulo requiere de un fichero de configuración: /etc/security/time.conf, que contiene reglas de la siguiente forma:

servicios ; ttys (terminales) ; usuarios ; fechas

Cada regla ocupa una línea, compuesta por los cuatro campos que acabamos de nombrar y separados por signos de punto y coma (;).

- servicios. Aquí indicamos una lista de servicios que serán afectados por esta regla.
- tty. Corresponde a una lista de nombres de terminales cubiertos por esta regla.
- usuario. Lista de nombres de usuarios a los que se aplica esta regla.

  Las listas de las que estamos hablando se refieren a una serie de objetos que además pueden relacionarse mediante operadores lógicos como el AND (&) o la negación (!). Un ejemplo de esto sería: jose&root.
- fecha. En este campo indicaríamos una lista con las fechas a las que esta regla se aplica. Los días se especifican como una secuencia de dos caracteres. Por ejemplo MoTuWe indican los días lunes, martes y miércoles. Los días de lunes a domingo vienen dados por las siguientes secuencias:

  Mo Tu We Th Fr Sa Su

También podemos especificar todos los días de la semana mediante Wk y los días del fin de semana, Wd.

Vemos como las secuencias corresponden a las primeras letras de los respectivos días en lengua inglesa.

pam_stack

En pocas palabras pam_stack permite llamar a pilas de módulos definidas para otros servicios. La idea es que tengamos un bloque común y si tenemos alguna vez que modificar todos los servicios sólo tengamos que cambiar dicho bloque común.

Componentes Auth, Account, Passwd y Session

Argumento reconocido: service=name

Descripción

Para cada componente añadimos una llamada al módulo pam_stack indicando el nombre del servicio cuya componente querramos usar. Así la siguiente línea

auth required pam_stack.so service=system-auth

utilizaría todos los módulos que apareciesen en la componente auth del fichero system-auth.

pam_xauth

pam_xauth está diseñado para intercambiar claves xauth (a veces conocidas como cookies) entre usuarios.

Sin pam_xauth, cuando xauth está habilitado y el usuario usa el comando su para asumir los privilegios de otro, este último no puede acceder al display del usuario original ya que no tiene las claves necesarias para el acceso.

Componente Session

Argumentos reconocidos: debug; xauthpath=

Descripción

pam_xauth resuelve el problema que acabamos de comentar llevando las claves desde el usuario que ejecuta su (el usuario fuente) hasta el usuario, cuya identidad está tomando el usuario fuente (usuario objetivo). Esto se hace cuando se crea la sesión y cuando se abandona se destruye la clave.

Esto significa, por ejemplo, que cuando ejecutamos su desde una sesión xterm podremos iniciar programas de las X sin tener que tratar explícitamente con xauth ni con los ficheros ~/Xauthority.

pam_xauth solo intercambiará claves si xauth puede listar una clave que esté conectada a la variable de entorno DISPLAY.

El control de acceso se realiza mediante dos ficheros; ~/xauth/export, en el directorio del usuario que invoca, y ~/xauth/import, en el directorio de usuario objetivo.

Si un usuario tiene un archivo ~/xauth/import sólo recibirá cookies de usuarios que aparezcan en dicho archivo. Si no existe podrá recibir cookies de cualquier usuario.

Si un usuario tiene un archivo ~/xauth/export sólo podrá intercambiar cookies con usuarios listados en ese archivo. Si no existe o el usuario que invoca no es root, entonces se intercambiarán cookies con cualquier otro, pero si es root no se intercambiarán con ningún otro usuario.

En cuanto a los argumentos, tenemos uno que habilita la creación de información de depuración y su envío a syslog. Es el debug.

El otro, xauthpath permite especificar otra ubicación para el programa xauth. Por defecto tenemos xauthpath=/usr/X11R6/bin/xauth.

pam_console

Pam_console está diseñado para dar a los usuarios en la consola física capacidades que de otro modo no tendrían, y por supuesto de eliminar dichas capacidades cuando ya no están en el sistema.

Las consolas pueden ser terminales virtuales o sesiones gestionadas por xdm, por defecto, aunque se pueden configurar.

En cuanto a las capacidades, este módulo provee de dos tipos: permisos de ficheros y autenticación.

Cuando un usuario entra al sistema mediante una consola y no hay ningún otro usuario, pam_console cambia los permisos y la propiedad de los dispositivos que se indican en /etc/security/console.perms, que sería el fichero de configuración de este módulo.

Ese usuario puede entonces acceder a diversos terminales que se considerarían parte de la primera consola. Poseerá dichos dispositivos hasta que no salga del último terminal que tenga abierto y será entonces cuando la consola quedará para el próximo usuario que acceda.

Los usuarios que hayan accedido a la consola mientras que el primero se encontraba en ella no tendrán la posesión de los elementos comentados.

Este módulo no es demasiado interesante para nosotros así que no vamos a entrar en más detalle.

pam_timestamp

Este módulo cachea los intentos de autenticación satisfactorios para usarlos como base de una posterior autenticación. Cuando una aplicación abre una sesión usando pam_timestamp, un archivo de marca de tiempo se crea en el directorio de usuario pertinente. Si pasado un tiempo, dentro de un margen especificado, ejecutamos de nuevo la aplicación, no volverá a pedir la contraseña y directamente accedemos a ella.

Componentes Auth y Session

Argumentos reconocidos: debug ; timestampdir= ; timestamp_timeout=N

Descripción

El argumento debug permite escribir logs a syslog para poder depurar el comportamiento de aplicación. También podemos especificar el directorio donde se va a escribir el fichero de marca de tiempo mediante timestampdir, y por supuesto, configurar el tiempo que dura la marca mediante timestamp_timeout.

Las marcas de tiempo de gestion conjuntamente con las de la aplicación sudo. Esto hace que se usen el mismo directorio para las marcas y el mismo margen de tiempo. Si no los indicamos se usaran las de sudo, que aparecen en el fichero sudoers, y si usamos dichos argumentos deben coincidir con los anteriores.

10.2 Componentes posibles de cada módulo

Módulo	Auth	Account	Passwd	Session
pam_unix	✓	✓	✓	✓
pam_cracklib	✗	✗	✓	✗
pam_deny	✓	✓	✓	✓
pam_env	✓	✗	✗	✗
pam_filter			✓	
pam_group	✓	✗	✗	✗
pam_lastlog	✗	✗	✗	✓
pam_limits	✗	✗	✗	✓
pam_nologin	✓	✗	✗	✗
pam_permit			✓	
pam_rootok	✓	✗	✗	✗
pam_securetty	✓	✗	✗	✗
pam_pwdb	✓	✓	✓	✓
pam_tally	✓	✓	✗	✗
pam_time	✓	✗	✗	✗
pam_stack	✓	✓	✓	✓
pam_xauth	✗	✗	✗	✓

Módulo	***Auth***	***Account***	***Passwd***	***Session***
pam_console	✓	✗	✗	✓
pam_timestamp	✓	✗	✗	✓

10.3 Creación de certificados digitales

Los certificados digitales son el equivalente digital del DNI, en lo que la autenticación de individuos se refieren. Permiten demostrar a un usuario que es quien dice ser, es decir, que está en posesión de la clave secreta asociada a su certificado.

En el caso que nos concierne, los certificados servirán para que los clientes verifiquen la identidad del servidor.

Antes de poder crear certificados debemos disponer de un certificado de autoridad certificadora (CA). Existen varias opciones para obtenerlo. Una de ellas es comprarlo. Diversas empresas, como por ejemplo verisign, los ofrecen por un precio razonable. Se consideran entidades de confianza y por tanto un certificado que provenga de ella tiene una amplia validez.

La otra opción es crear nuestro propio certificado de CA usando para ello OpenSSL. Esta será la elegida puesto que cumple con nuestras necesidades y además es gratuita.

Los pasos para conseguir los certificados necesarios los presentamos a continuación.

1) Creación de un certificado de CA. Para ello debemos contar con directorio reservado para albergar algunos ficheros, y hacer uso del script CA.sh que proporciona OpenSSL.

```
mkdir /var/miCA
cd /var/miCA
/usr/local/openssl/misc/CA.sh -newca
```

Tras este último debemos contestar las perguntas que se nos formulan.

```
CA certificate filename (or enter to create)     [pulsamos enter]

Making CA certificate ...
Generating a 1024 bit RSA private key
............++++++
..........++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:                              [introducimos una
clave]
Verifying - Enter PEM pass phrase:              [la reintroducimos]
```

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]: ES                    [código del pais]
State or Province Name (full name) [Some-State]:Sevilla [provincia]
Organization Name (eg, company) [Internet Widgits Pty Ltd]: cdc [nombre]
Organizational Unit Name (eg, section) []:. [descripción]
Common Name (eg, YOUR name) []:gotche                    [nombre del servidor]
Email Address []:j.martin.bejarano@gmail.com             [correo de referencia]
```

Prácticamente todos los campos son descriptivos, pero con uno debemos tener especial cuidado. Este es “Common Name”. En él debemos especificar el FQDN de la máquina donde reside el servidor, en nuestro caso OpenLDAP.

Es muy importante que sea el mismo ya que sino el certificado no se corresponderá con el servidor y no será aceptado, imposibilitando por tanto, la conexión segura.

El FQDN del equipo donde se encuentra el servidor se averigua mediante el siguiente comando.

```
hostname –f
```

2) El siguiente paso es crear una petición de certificado y una clave privada para el servidor. Para ello debemos ejecuta el siguiente comando. Debemos tener en cuenta que OpenLDAP no trabaja con claves privadas encriptadas, por ello la aparición de la opción -nodes.

```
openssl req –new –nodes –keyout newreq.pem –out newreq.pem
```

Se nos presenta un menú interactivo y al igual que en el caso anterior debemos contestar a todas las cuestiones que se nos formulan.

```
Generating a 1024 bit RSA private key
..........++++++
.....................................................................
.........++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Sevilla
Locality Name (eg, city) []:Sevilla
Organization Name (eg, company) [Internet Widgits Pty Ltd]: cdc
Organizational Unit Name (eg, section) []:.
Common Name (eg, YOUR name) []:gotche
Email Address []:j.martin.bejarano@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:                                    [clave
extra]
An optional company name []:.                          [información
extra]
```

3) Para que el certificado tenga validez debe ser firmado por la CA. Para ello:

```
/usr/local/openssl/misc/CA.sh -sign
```

Pedirá la clave que indicamos al principio. Debe ser la misma, sino no nos permitirá seguir.

Puede darse el caso de que obtengamos una salida errónea al ejecutar el comando anterior. Esto es debido a que el fichero de configuración posee una opción de seguridad activada por defecto. Se trata de la opción “unique_subject” y no permite crear distintos certificados con el

mismo asunto. Debemos eliminar el comentario y dejarla de esta forma.

```
unique_subject  =  no
```

Durante el firmado del certificado se nos pedirá que aceptemos las operaciones y a continuación se presentará por pantalla el certificado.

```
Using configuration from /usr/local/openssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Mar 26 15:57:31 2005 GMT
            Not After : Mar 26 15:57:31 2006 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Sevilla
            localityName              = Sevilla
            organizationName          = cdc
            commonName                = gotche
            emailAddress              = j.martin.bejarano@gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                88:6E:5B:02:3C:3D:6F:E5:D7:10:92:41:24:48:F5:2B:19:71:
24:C5
            X509v3 Authority Key Identifier:
          keyid:20:E3:0E:68:F0:19:EF:1F:AA:2C:14:7F:01:A7:38:39:A1:42:
4F:15
            DirName:/C=ES/ST=Sevilla/L=Sevilla/O=cdc
            /CN=gotche/email Address=j.martin.bejarano@gmail.com
                serial:00
Certificate is to be certified until Mar 26 15:57:31 2006 GMT (365
days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Certificate:
```

```
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=ES, ST=Sevilla, L=Sevilla, O=cdc,
        CN=gotche/emailAddress=j.martin.bejarano@gmail.com
        Validity
            Not Before: Mar 26 15:57:31 2005 GMT
            Not After : Mar 26 15:57:31 2006 GMT
        Subject: C=ES, ST=Sevilla, L=Sevilla, O=cdc,
        CN=gotche/emailAddress=j.martin.bejarano@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:cc:65:35:ec:4a:f8:de:fe:4c:2e:66:63:ca:e5:
                    90:ca:44:ed:0f:4f:a1:50:a7:92:99:89:51:0e:0c:
                    76:fb:55:68:31:c1:3f:b2:fd:ff:90:dc:b8:66:ce:
                    ed:3a:e9:b0:c8:60:a4:99:66:7e:86:8d:78:c3:6d:
                    ba:9d:46:57:5b:69:6c:3c:2b:f2:2d:c4:2c:2f:d6:
                    66:b0:14:9c:8e:15:9a:7d:32:45:87:e8:d3:46:9b:
                    41:04:f1:bc:3c:1a:03:a8:1d:94:8d:ae:94:7a:8a:
                    cc:58:d0:f6:fb:4f:c2:21:5a:58:28:88:cb:fc:59:
                    28:6f:cb:d3:3f:a8:5b:06:a5

                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                88:6E:5B:02:3C:3D:6F:E5:D7:10:92:41:24:48:F5:2B:19:71:
24:C5
            X509v3 Authority Key Identifier:
              keyid:20:E3:0E:68:F0:19:EF:1F:AA:2C:14:7F:01:A7:38:39:A1:42:
4F:15
                DirName:/C=ES/ST=Sevilla/L=Sevilla/O=cdc
                /CN=gotche/emailAddress=j.martin.bejarano@gmail.com
                serial:00

    Signature Algorithm: md5WithRSAEncryption
        b3:7b:9f:e4:34:91:f9:b1:7e:37:1d:14:ea:38:85:ac:6b:f9:
        f9:cc:db:c5:92:76:ff:fc:cc:2c:67:a3:81:80:6c:aa:f1:a5:
```

```
        15:23:9d:38:09:f5:1e:99:c2:a6:8d:00:f7:c7:61:04:1c:5c:
        b2:dc:df:5f:41:fc:33:f2:42:e8:5c:d4:4a:25:b5:e8:9d:6f:
        62:da:cb:db:90:8b:f7:fa:66:94:4f:21:f2:3b:9f:46:70:c4:
        39:cf:f9:8e:03:e5:dc:83:6a:d3:c2:e1:c1:16:0d:ad:24:79:
        a7:9f:af:a0:cc:1c:c0:03:42:22:07:97:e8:0b:13:3a:90:89:
        9a:63
-----BEGIN CERTIFICATE-----
MIIDtjCCAx+gAwIBAgIBATANBgkqhkiG9w0BAQQFADCBkDELMAkGA1UEBhMCRVMx
EDAOBgNVBAgTB1NldmlsbGExEDAOBgNVBAcTB1NldmlsbGExGjAYBgNVBAoTEUNl
bnRybyBkZSBDYWxjdWxvMRUwEwYDVQQDEwxnb3RjaGUubG9jYWwxKjAoBgkqhkiG
9w0BCQEWG2oubWFydGluLmJlamFyYW5vQGdtYWlsLmNvbTAeFw0wNTAzMjYxNTU3
MzFaFw0wNjAzMjYxNTU3MzFaMIGQMQswCQYDVQQGEwJFUzEQMA4GA1UECBMHU2V2
aWxsYTEQMA4GA1UEBxMHU2V2aWxsYTEaMBgGA1UEChMRQ2VudHJvIGRlIENhbGN1
bG8xFTATBgNVBAMTDGdvdGNoZS5sb2NhbDEqMCgGCSqGSIb3DQEJARYbai5tYXJ0
aW4uYmVqYXJhbm9AZ21haWwuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDMZTXsSvje/kwuZmPK5ZDKRO0PT6FQp5KZiVEODHb7VWgxwT+y/f+Q3Lhmzu06
6bDIYKSZZn6GjXjDbbqdRldbaWw8K/ItxCwv1mawFJyOFZp9MkWH6NNGm0EE8bw8
GgOoHZSNrpR6isxY0Pb7T8IhWlgoiMv8WShvy9M/qFsGpQIDAQABo4IBHDCCARgw
CQYDVR0TBAIwADAsBglghkgBhvhCAQ0EHxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2Vy
dGlmaWNhdGUwHQYDVR0OBBYEFIhuWwI8PW/l1xCSQSRI9SsZcSTFMIG9BgNVHSME
gbUwgbKAFCDjDmjwGe8fqiwUfwGnODmhQk8VoYGWpIGTMIGQMQswCQYDVQQGEwJF
UzEQMA4GA1UECBMHU2V2aWxsYTEQMA4GA1UEBxMHU2V2aWxsYTEaMBgGA1UEChMR
Q2VudHJvIGRlIENhbGN1bG8xFTATBgNVBAMTDGdvdGNoZS5sb2NhbDEqMCgGCSqG
SIb3DQEJARYbai5tYXJ0aW4uYmVqYXJhbm9AZ21haWwuY29tggEAMA0GCSqGSIb3
DQEBBAUAA4GBALN7n+Q0kfmxfjcdFOo4haxr+fnM28WSdv/8zCxno4GAbKrxpRUj
nTgJ9R6ZwqaNAPfHYQQcXLLc319B/DPyQuhc1Eolteidb2Lay9uQi/f6ZpRPIfI7
n0ZwxDnP+Y4D5dyDatPC4cEWDa0keaefr6DMHMADQiIHl+gLEzqQiZpj
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
```

4) Además de obtener la salida por pantalla, disponemos de unos ficheros que contienen los certificados y la clave privada del servidor. Deben ser copiados en una localización accesible por el servidor.

```
cp /var/miCA/demoCA/cacert.pem /usr/local/etc/openldap
mv newcert.pem /usr/local/etc/openldap/servercrt.pem
mv newreq.pem /usr/local/etc/openldap/serverkey.pem
chmod 600 /usr/local/etc/openldap/serverkey.pem
```

Mediante la última línea protegemos la clave privada ya que, como dijimos al principio, no está encriptada. OpenLDAP no soporta que lo esté.

5) A partir de este momento ya se encuentra todo preparado para que el servidor haga uso de los certificados. Sólo quedaría por establecer la

configuración del servidor y del cliente para que hagan uso de ellos.

En el caso del servidor habría que añadir al fichero slapd.conf las siguientes líneas.

```
TLSCACertificateFile /usr/local/etc/openldap/cacert.pem
TLSCertificateFile /usr/local/etc/openldap/servercrt.pem
TLSCertificateKeyFile /usr/local/etc/openldap/serverkey.pem
```

En los clientes tendrían que añadir al fichero de configuración ldap.conf la siguiente línea.

```
TLS_CACERT /etc/openldap/cacert.pem
```

Dependiendo de la distribución que usemos, ldap.conf se puede encontrar en una localización o en otra. Incluso puede haber varios. De hecho pueden existir diferentes programas que accedan al servidor y tengan diferentes ficheros de configuración. Se deben configurar todos para que no queden aplicaciones sin poder usar el servidor de forma segura.

También debemos copiar el certificado de CA al cliente, en una localización adecuada.

10.4 Script qmail.sh

```
#!/bin/bash
#
# Autor: Jose A. Martin
#
# Descripcion: Instalacion de qmail
# Contamos con los usuario necesarios ya creados (qmaill,qmaild,...)
# Haremos uso de los paquetes
#
#  * netqmail
#  * ucspi-tcp
#  * daemontools

# Parametros de configuracion

# Nombre de la maquina servidor
HOST=gotche

# Alias para los usuarios root y postmaster
ALIAS_ROOT=jose
ALIAS_POSTMASTER=jose

# Instalacion
umask 022
mkdir -p                      /package
cp daemontools-0.76.tar.gz    /package
chmod 1755                    /package

# Desempaquetamos, parcheamos y configuramos (qmail)
cd /usr/local/src
gunzip netqmail-1.05.tar.gz
tar xpf netqmail-1.05.tar
cd netqmail-1.05
gunzip -c qmail-1.03.tar.gz | tar xf -
cd qmail-1.03
patch -p1 < /usr/local/src/qmail/parches/qmail-ldap-1.03-20041201.patch
cp -f /usr/local/src/qmail/Makefile-ldap Makefile
cd ../..

# Desempaquetamos  (ucspi y daemontools)
gunzip      ucspi-tcp-0.88.tar.gz
tar xpf     ucspi-tcp-0.88.tar
rm          ucspi-tcp-0.88.tar
cd          /package
gunzip      daemontools-0.76.tar.gz
tar xpf     daemontools-0.76.tar
rm          daemontools-0.76.tar

# Creamos el directorio para qmail
mkdir -p    /var/qmail

# Construimos los paquetes
cd /usr/local/src/netqmail-1.05/qmail-1.03
make setup check && echo "Todo ok"

# Configuracion
# Debemos ejecutar config para que halle el nombre del servidor en el dns
# aunque por ahora vamos a ejecutarlo de forma local.
#  ./config
./config-fast $HOST

# Instalacion de ucspi-tcp parcheada
```

```
cd /usr/local/src/ucspi-tcp-0.88
patch < /usr/local/src/qmail/parches/ucspi-tcp-0.88.errno.patch
make
make setup check

# Instalacion de daemontools
cd /package/admin/daemontools-0.76
cd src
patch < /usr/local/src/qmail/parches/daemontools-0.76.errno.patch
cd ..
package/install

# Creamos el script de arranque de qmail rc
cp          /usr/local/src/rc                   /var/qmail/
chmod 755   /var/qmail/rc
mkdir -p    /var/log/qmail

# Elegimos un directorio Maildir para contener los mensajes
echo ./Mailbox./ >/var/qmail/control/defaultdelivery

# Creamos el script qmailctl para el inicio automatico de qmail cuando
# el sistema arranca y cuando se apague que lo haga correctamente
cp          /usr/local/src/qmail/scripts/qmailctl    /var/qmail/bin/
ln -s       /var/qmail/bin/qmailctl                  /usr/bin

# Creamos varios scripts que nos serviran para supervisar qmail
mkdir -p /var/qmail/supervise/qmail-send/log
mkdir -p /var/qmail/supervise/qmail-smtpd/log

cp /usr/local/src/qmail/scripts/send-run       /var/qmail/supervise/qmail-
send/run
cp /usr/local/src/qmail/scripts/send-log-run   /var/qmail/supervise/qmail-
send/log/run

cp /usr/local/src/qmail/scripts/smtpd-run      /var/qmail/supervise/qmail-
smtpd/run
cp /usr/local/src/qmail/scripts/smtpd-log-run  /var/qmail/supervise/qmail-
smtpd/log/run

# Establecemos a 20 los mensajes que puede tratar simultaneamente
echo 20 > /var/qmail/control/concurrencyincoming
chmod 644 /var/qmail/control/concurrencyincoming

# Establecemos los directorios para los logs
mkdir -p /var/log/qmail/smtpd
chown qmaill /var/log/qmail                    /var/log/qmail/smtpd

# Seran tratados por las daemontools. Para ello deben estar en /service
ln -s       /var/qmail/supervise/qmail-send    /var/qmail/supervise/qmail-
smtpd /service

# Esperamos unos momentos para que el sistema qmail arranque
sleep 10

# y apagamos
qmailctl stop

# Permitimos mandar correo a localhost
echo '127.:allow,RELAYCLIENT=""' >>/etc/tcp.smtp
qmailctl cdb

# Configuracion para encontrar el servidor OpenLDAP
echo 192.168.123.100 > /var/qmail/control/ldapserver
echo "dc=example,dc=com" > /var/qmail/control/ldapbasedn
```

```
# Reemplazamos los posibles enlaces de sendmail
# No pasa nada si no existen
mv /usr/lib/sendmail    /usr/lib/sendmail.old   || echo "No existe
ejecutable sendmail"
mv /usr/sbin/sendmail   /usr/sbin/sendmail.old  || echo "No existe
ejecutable sendmail"

touch       /usr/lib/sendmail.old
touch       /usr/sbin/sendmail.old
chmod 0     /usr/lib/sendmail.old               /usr/sbin/sendmail.old
ln -s       /var/qmail/bin/sendmail             /usr/lib
ln -s       /var/qmail/bin/sendmail             /usr/sbin

# Por ultimo creamos algunos alias necesarios

echo $ALIAS_ROOT >                          /var/qmail/alias/.qmail-root
echo $ALIAS_POSTMASTER >                    /var/qmail/alias/.qmail-postmaster
ln -s /var/qmail/alias/.qmail-postmaster        /var/qmail/alias/.qmail-
mailer-daemon
chmod 644 /var/qmail/alias/.qmail-root          /var/qmail/alias/.qmail-
postmaster

# Reiniciamos qmail y listo
qmailctl start

# Fin del script de instalacion
```

10.5 Bibliografía

[1] **LDAP Programming, Management and Integration**

Clayton Donley

Manning Publications, 2003

[2] **Understanding and Deploying LDAP Directory Services, Second Edition**

Timothy A. Howes Ph.D., Mark C. Smith, Gordon S. Good

Addison Wesley, 2003

[3] **LDAP Programming with JAVA**

Rob Weltman, Tony Dahbura

Addison Wesley, 2000

[4] **The ABCs of LDAP: How to Install, Run, and Administer LDAP Services**

Reinhard Voglmaier

Auerbach Publications, 2004

[5] **Guía de instalación de OpenLDAP (Administrator Guide 2.2)**

www.openldap.org

[6] **LDAP System Administration**

Gerald Carter

O'Reilly, 2003

[7] **LDAP implementation Cookbook**

Heinz Johner, Michel Melot, ...

IBM, 1999

[8] **Autenticación Unix**

http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html

[9] **User authentication and encryption overview**
Eric Guerrino, Mike Kahn, and Ellen Kapito
1997

[10] **Proyecto Linux-PAM**
www.kernel.org/pub/linux/libs/pam

[11] **Simple Authentication and Security Layer (SASL)**
www.faqs.org/rfcs/rfc2222.html (RFC 2222)
http://asg.web.cmu.edu/sasl/sasl-library.html (Página oficial)

[12] **Practical Unix & Internet Security, 3rd Edition**
By Simson Garfinkel, Alan Schwartz, Gene Spafford
O'Reilly, 2003

[13] **Linux Security Cookbook**
Daniel J. Barret, Robert G. Byrnes, Richard Silverman
O'Reilly 2003

[14] **Red Hat Linux 9.0: Manual oficial de referencia de Red Hat Linux**
www.redhat.com

[15] **Learning Red Hat Linux 3rd Edition**
Bill McCarthy
O'Reilly, 2003

[16] **Squid: The Definitive Guide**
Duane Wessels
O'Reilly, 2004

[17] **Documentación de Squid**
www.squid-cache.org

[18] **Listas negras para squid**
www.squidguard.org/blacklist

[19] **Documentación de PostgreSQL**
www.postgresql.org

[20] **Tuning PostgreSQL for performance**
Shridhar Daithankar, Josh Berkus
2003

[21] **Documentación de qmail**
www.qmail.org

[22] **qmail**
John Levine
O'Reilly, 2004

[23] **The qmail Handbook**
Dave Sill
Apress, 2002

[24] **Documentación de SqWebMail**
http://www.courier-mta.org/sqwebmail/