Autenticación usando PAM

Sea cual sea el sistema operativo usado, múltiples aplicaciones y servicios necesitan una identificación y una autenticación del usuario, es decir, comprobar que el usuario está autorizado a acceder a un servicio o a una aplicación, y que este usuario prueba su identidad.

Al principio, en entorno Unix, las aplicaciones y los servicios desarrollaban su propio método de identificación y de autentificación del usuario y de control de acceso. Algunos programas validaban la cuenta del usuario a partir de la base de cuentas local (`/etc/passwd`) y otros a partir de una base de cuentas distribuida (NIS, LDAP). Algunas aplicaciones comprobaban que la cuenta que presentaba el usuario no era la cuenta del superusuario, o al contrario que se trataba del superusuario, otras aplicaciones daban el acceso a los usuarios según que existiera o no el archivo (`/etc/nologin`, por ejemplo), o la declaración de su terminal en otro archivo (`/etc/securettys`), etc.

En las primeras versiones Unix, la contraseña estaba almacenada directamente en la línea de la cuenta del usuario, cifrada, en el archivo `/etc/passwd`. No obstante, como todos los usuarios debían poder acceder a este archivo en lectura, se creó un archivo específico de gestión de contraseñas, `/etc/shadow`, con funcionalidades suplementarias (duración mínima, máxima, etc.). Es el método implementado en Linux.

Para simplificar y racionalizar los métodos de control de acceso de las aplicaciones, se concibió una capa intermediaria, que permite a las aplicaciones y los servicios delegarle estas operaciones: PAM (*Pluggable Authentication Modules*). Creada por Unix, ha sido implementada en Linux.

PAM está compuesta por un conjunto de módulos de software especializados, que podemos combinar en un orden específico y según reglas precisas, a través de archivos de configuración que son asociables a las diferentes aplicaciones.

Este método permite a los administradores de sistemas configurar las reglas de control de acceso a los servicios que utilizan PAM, independientemente de los propios programas, para adaptarlos a la política de seguridad de su organización.

1. El principio

PAM es una capa de software que asegura la interfaz entre las aplicaciones y los diversos métodos de identificación, autentificación y de control de acceso disponibles en el sistema.

Una aplicación compatible con PAM no administra los procedimientos de identificación, autentificación y control de acceso del usuario ella misma. Delegará en la biblioteca PAM este conjunto de operaciones. La capa PAM, en función del archivo de configuración asociado a la aplicación y definido por el administrador de sistemas, ejecutará sucesivamente un conjunto de módulos para comprobar la identidad del usuario y determinar si está habilitado para acceder a la aplicación. La capa PAM devuelve a la aplicación el resultado del control, positivo o negativo.

La capa PAM dispone de un gran número de módulos autónomos (se trata de módulos objeto que se pueden compartir, de tipo `.so`), que pueden estar apilados y ser ejecutados en un orden preciso. La elección y el orden de ejecución de los módulos, la estrategia de pasar de un módulo a otro o de salir del procedimiento de control (positivo o negativo), depende del archivo de configuración asociado a la aplicación.

2. Los módulos PAM

Los módulos PAM que se pueden usar en un sistema Linux son muy numerosos (algunas decenas) y de uso variado. Algunos son muy utilizados, otros son más específicos.

a. Los tipos de acción de PAM

PAM administra cuatro tipos de operaciones, para los servicios o las aplicaciones. La aplicación que hace una llamada a la biblioteca PAM especifica el tipo de acción que se tiene que efectuar. PAM ejecuta el o los módulos configurados y que corresponden a esta acción. Devuelve un código de autorización (éxito) o de rechazo (fallo) a la aplicación.

Los tipos de acciones son las siguientes:

- `account` : gestión de la identidad del usuario. Los módulos consiguen la identidad del usuario y se aseguran de que está autorizado a acceder a la aplicación.

- `auth`: los módulos de autentificación controlan la identidad efectiva del usuario (control de la validez de los elementos de autentificación proporcionados, contraseña o de otro tipo).
- `session`: configuración de la apertura de la sesión (variables de entorno, límites de recursos, etc.).
- `password`: gestión de la contraseña (número de caracteres, historial, respeto de las reglas de complejidad, etc.).

b. Las pilas de módulos

Los módulos PAM son conectables, se pueden ejecutar sucesivamente, bajo la forma de una pila de módulos. Durante el procedimiento de autentificación, la capa PAM ejecuta sucesivamente cada módulo de la pila, en el orden de declaración que se encuentra en el archivo de configuración.

Cada módulo devuelve un código, que puede ser positivo o negativo. El paso al módulo siguiente depende del código devuelto por el módulo y de la regla de prioridad asociada al módulo (obligatoria, opcional, suficiente, etc.).

Para una acción en concreto, la autentificación por ejemplo, pueden ser llamados distintos módulos PAM. En ese caso se dirá que nos encontramos ante una pila de módulos PAM. El funcionamiento en modo de pila es una de las mayores contribuciones de los servicios PAM.

c. Los módulos principales PAM

Los módulos se encuentran en archivos de módulos objetos que se pueden compartir (`.so`), cuya ubicación por defecto es el directorio `security`, en la arborescencia `/usr/lib*`.

Los módulos más utilizados son los siguientes:

`pam_unix.so`	Autenticación usando los archivos de cuenta de usuarios locales `/etc/passwd y /etc/shadow`.

`pam_securetty.so`	La apertura de sesión con la cuenta de superusuario solamente será aceptada por los terminales que se encuentren listados en `/etc/securetty` .
`pam_nologin.so`	Si el archivo `/etc/nologin` existe, solamente podrá abrir una sesión el superusuario.
`pam_listfile.so`	Aplica reglas de control de acceso definidas en el archivo usado como argumento del módulo.
`pam_limits.so`	Aplica limitaciones de recursos a usuarios o a grupos, definidos en el archivo `/etc/security/limits.conf` .
`pam_cracklib.so`	Controla el nivel de seguridad de una contraseña.
`pam_selinux.so`	Si SELinux está activado, controla el contexto de seguridad del terminal de conexión.
`pam_sss.so`	Módulo de interfaz con el daemon `SSSD` (*System Security Services Daemon*).
`pam_deny.so`	Rechaza la conexión. Se ejecuta generalmente si ningún otro módulo ha conseguido autenticar al usuario.
`pam_permit.so`	Autoriza la conexión. Se ejecuta generalmente si ningún otro módulo ha fallado en la autenticación del usuario.
`pam_env.so`	Declara variables de entorno leídas en `/etc/environment` o en el archivo al que se hace referencia con el parámetro `envfile=` .

`pam_lastlog.so`	Muestra los datos de la última apertura de sesión exitosa.
`pam_mail.so`	Comprueba la presencia de nuevos mensajes en la cuenta de mensajería local del usuario.

3. Configuración de PAM

Las primeras versiones de PAM tenían como archivo de configuración `/etc/pam.conf`. La generalización del uso de PAM y el aumento del número de módulos ha hecho necesario que se distribuyan las configuraciones en archivos independientes. En la mayoría de las distribuciones recientes, los archivos de configuración de PAM se encuentran en el directorio `/etc/pam.d` (si este directorio existe, se ignorará el archivo `/etc/pam.conf`).

Ejemplo

Contenido del directorio `/etc/pam.d` para una distribución CentOS 8:

```
ls /etc/pam.d
atd             crond           gdm-launch-environment  login           postlogin
smartcard-auth  sudo            vlock
chfn            cups            gdm-password            other           remote
squid           sudo-i          vmtoolsd
chsh            fingerprint-auth  gdm-pin               passwd          runuser
sshd            su-l            xserver
cockpit         gdm-autologin   gdm-smartcard           password-auth   runuser-l
sssd-shadowutils  system-auth
config-util     gdm-fingerprint  liveinst               polkit-1        samba
su              systemd-user
```

Contenido del directorio `/etc/pam.d` para una distribución Debian 10:

```
ls /etc/pam.d
chfn            common-auth                 cron                    gdm-password
passwd          runuser-l                   sudo
chpasswd        common-password             gdm-autologin           login
polkit-1        samba                       su-l
chsh            common-session              gdm-fingerprint         newusers
ppp             sshd                        systemd-user
common-account  common-session-noninteractive  gdm-launch-environment  other
runuser         su
```

Los archivos de configuración tienen nombres derivados de las aplicaciones correspondientes (`sshd`, `login`, `passwd`, etc.).

a. Estructura de los archivos de configuración

Cada aplicación que usa PAM está asociada a un archivo de configuración en el directorio `/etc/pam.d`.

Se trata de un archivo de texto; cada línea útil describe un módulo PAM y su configuración, según el formato siguiente:

```
TipoAcción TipoCtrl|Include Módulo|ArchivoPAM [Args]
```

donde:

`TipoAcción`	Tipo de acción PAM (`auth`, `account`, `session` o `password`).
`TipoCtrl`	Tipo de control según el resultado del módulo (`required`, `requisite`, `sufficient` u `optional`).
`Módulo`	Nombre del módulo PAM.
`Include ArchivoPAM`	Línea de inclusión de otro archivo de configuración PAM.
`Args`	Argumentos que se pasarán al módulo (opcional).

El orden de las líneas de configuración de los módulos determina su orden de ejecución.

Cuando la biblioteca PAM recibe una solicitud de parte de la aplicación, esta ejecuta el primero de los módulos que corresponde al tipo de la solicitud (tipo de acción). Este último devuelve un valor después de su ejecución, que solo puede ser éxito o fallo. Según el resultado del módulo y el tipo de control que se le ha asociado, la biblioteca pasará o no al módulo siguiente de la pila.

Cuando la pila de los módulos ejecutados termina en un resultado definitivo, la biblioteca PAM devuelve a la aplicación: éxito o fallo en el procedimiento de autentificación.

b. Encadenamiento de los módulos

Para una aplicación y un tipo de acción dado, el encadenamiento de los módulos está determinado por el parámetro de control (*control flag*) de cada módulo ejecutado.

Este parámetro, obligatorio, puede presentar los valores siguientes:

- `required`: el módulo tiene que devolver obligatoriamente un código de éxito. Si falla, los módulos siguientes de la pila no se ejecutarán y la acción devolverá un código de fallo.

- `requisite` : el módulo tiene que devolver obligatoriamente un código de éxito. Si falla, los módulos siguientes de la pila serán ejecutados, pero la acción devolverá un código de fallo.
- `sufficient` : si el módulo devuelve un código de éxito y si ningún módulo `requisite` anterior ha fallado, los módulos siguientes de la pila no son ejecutados, la acción devuelve un código de éxito.
- `optional` : el código devuelto por el módulo será ignorado, excepto si se trata del único módulo de la pila, en este caso la acción devolverá su código de retorno.

c. Ejemplo

Numerosas aplicaciones estándar se basan en la biblioteca PAM. Vamos a usar `sshd` para explicar el funcionamiento de los módulos con un ejemplo.

El archivo de configuración PAM usado por el daemon `sshd` es `/etc/pam.d/sshd` . He aquí su contenido por defecto en una distribución CentOS 8:

```
auth       substack     password-auth
auth       include      postlogin
account    required     pam_sepermit.so
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed
in the user context
session    required     pam_selinux.so open env_params
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    optional     pam_motd.so
session    include      password-auth
session    include      postlogin
```

Para el tipo de acción `account` (control de la cuenta de usuario), encontramos dos

módulos `required`, y la inclusión del archivo de configuración general `password-auth`.

El primer módulo obligatorio es sobre SELinux. El siguiente prohíbe la conexión a los usuarios cuyo UID es diferente de `0`, si hay un archivo `/etc/nologin` que existe.

Podemos comprobar el efecto de ese módulo creando el archivo:

```
> /etc/nologin
```

Después, probamos la conexión al servidor local `sshd` con una cuenta de no superusuario:

```
ssh -l pba localhost
Bienvenido al servidor CentOS8.
El acceso está reservado a personas autorizadas.
pba@localhost's password:
Connection closed by::1 port 22
```

La conexión ha sido denegada.

Modificamos el archivo de configuración PAM de `sshd`, poniendo en comentario la línea del módulo `pam_nologin.so`.

Volvemos a lanzar de nuevo el comando de conexión:

```
ssh -l pba localhost
Bienvenido al servidor CentOS8.
El acceso está reservado a personas autorizadas.
pba@localhost's password:
Bienvenido al servidor CentOS 8
Last failed login: Thu Jun  4 13:20:27 CEST 2020 from::1 on ssh:notty
There were 7 failed login attempts since the last successful login.
Last login: Thu Jun  4 09:16:14 2020
```

Esta vez, la conexión ha sido aceptada.

4. ConfiguraciónPAM para SSSD

SSSD (*System Security Services Daemon*) es un servicio de software que permite gestionar diferentes mecanismos de autentificación de los usuarios, usando fuentes remotas (LDAP, NIS, Samba, Kerberos, etc.), de manera transparente para las aplicaciones.

Su conexión con las aplicaciones se hace gracias a la configuración NSS (*Name Service Switch*) y usando un módulo PAM.

a. Configuración NSS

Una vez que el servicio SSSD, administrado por el daemon `sssd`, esté instalado y configurado (archivo `/etc/sssd/sssd.conf`), el uso de este servicio para la autenticación tiene que ser especificado en el archivo de configuración NSS: `/etc/nsswitch.conf` .

Este archivo de texto contiene una línea por cada tipo de servicio (nombre de host, contraseñas, cuentas de grupos, etc.). Cada línea define cuáles son las fuentes que se tienen que usar para obtener el tipo de información y en qué orden hay que interrogarlas.

Para el procedimiento de autentificación de un usuario, se tienen que configurar cuatro tipos de servicios: la cuenta de usuario (`passwd`), la gestión de la contraseña del usuario (`shadow`), los grupos de usuarios locales (`group`) y, si fuera necesario, los grupos de usuarios de red (`netgroup`).

La palabra clave para la fuente `SSSD` es `sss`.

Ejemplo

```
passwd: files sss
group: files sss
shadow: files sss
```

En este ejemplo, la autenticación y la gestión de la contraseña se hacen primero mediante los archivos locales (`/etc/passwd` , `/etc/group` o `/etc/shadow`) y, si el usuario no se encuentra, mediante una llamada al daemon `sssd`.

b. Configuración PAM

Los procedimientos de identificación y de autentificación de una cuenta de usuario y de gestión de su contraseña están generalmente definidos en dos archivos de configuración de PAM, incluidos si fuera necesario con la directiva `include` en los archivos de configuración PAM de las aplicaciones: `/etc/pam.d/system-auth` y `/etc/pam.d/password-auth` para las distribuciones de tipo Red Hat, `/etc/pam.d/common-password` y `/etc/pam.d/common-auth` para las distribuciones de tipo Debian.

El módulo `pam_sss.so` tiene que estar declarado en los archivos de configuración, para el tipo de acción `auth` o `password`. El tipo de control es generalmente `sufficient`, y el módulo toma un argumento que puede ser, entre otros, `use_first_pass` (evita que el usuario tenga que teclear de nuevo la contraseña) o `use_authtok` (considera que la contraseña ha sido validada por un módulo anterior).

Ejemplo

En el archivo de configuración `/etc/pam.d/system-auth` de una distribución CentOS 8 donde `SSSD` está instalado:

```
[...]
auth        sufficient    pam_sss.so use_first_pass
[...]
```

En el archivo de configuración `/etc/pam.d/common-password` de una distribución Debian 10 donde `SSSD` está instalado:

```
password    sufficient              pam_sss.so use_authtok
```