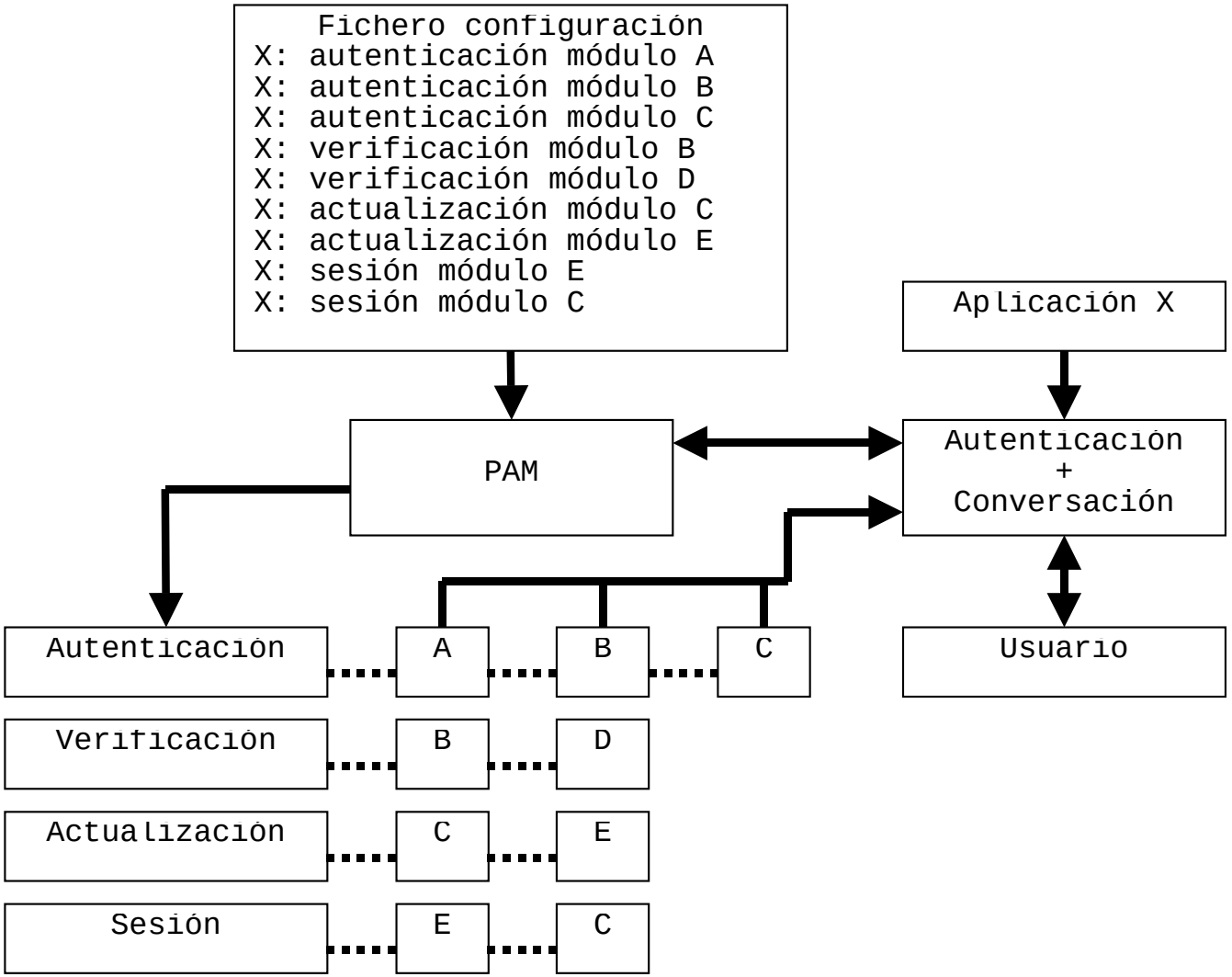
Introducción (I)

- Pluggable Authentication Module (PAM) son un conjunto de librerías que permiten:
  - Seleccionar como los usuarios se identifican ante una aplicación.
  - Cambiar el funcionamiento de un método de autenticación sin necesidad de cambiar las aplicaciones que lo utilizan.
  - Cambiar el método de autenticación de una aplicación cambiando las librerías de PAM usadas.

Introducción (II)



Configuración de PAM (I)

- Dos métodos de configuración de la autenticación de una aplicación:
  - Fichero /etc/pam.conf (obsoleto).
  - Ficheros del directorio /etc/pam.d
- Si se encuentra en /etc/pam.d no se utiliza el de pam.conf.
- Sintaxis de los ficheros:
  - Líneas que comienzan por # son comentarios.
  - Línea que termina con \ continúa en la línea siguiente.

Configuración de PAM (II)

- Las líneas de configuración tienen la sintaxis:
  - Para líneas del fichero /etc/pam.conf:

```
servicio   tipo   control   camino al módulo   argumentos del módulo
```

  - Para líneas de ficheros del directorio /etc/pam.d:

```
tipo   control   camino al módulo   argumentos del módulo
```

- En el segundo caso, el fichero debe tener como nombre “servicio”.

El campo servicio

- Indica el nombre del servicio que utiliza este método de autenticación.
- Suele coincidir con el nombre del servidor (vsftpd, sshd, etc.).
- Existe un método especial, llamado OTHER, que se utiliza como método por defecto en caso de que no exista uno particular para el servicio.

El campo tipo

- Especifica uno de los cuatro tipos de módulos existentes:
  - auth (autenticación): Autentica al usuario mediante un método y le proporciona los privilegios.
  - account (verificación): Comprueba si el usuario tiene permiso para utilizar el servicio o si el servicio esta permitido (control horario, por ejemplo).
  - password (actualización): Actualiza el mecanismo de autenticación.
  - session (sesión): Acciones que deben ejecutarse antes y/o después del acceso del usuario.

El campo control (I)

- Indica como debe actuar PAM ante un fallo en las condiciones del módulo.
  - Ignorar el fallo.
  - Terminar la ejecución.
  - Etc.
- Una aplicación siempre recibe una respuesta sumaria del tipo correcto o incorrecto, sin que sepa, en el segundo caso, que ha provocado el fallo.

El campo control (II)

- Existen dos sintaxis:
  - Simple o histórica.
  - Nueva.
- La simple es una palabra que define la severidad asociada al cumplimiento o no del módulo.
- Existen cuatro palabras:
  - requisite: Se envía fallo sin ejecutar otros módulos.
  - required: Se enviará fallo, ejecutando previamente los otros módulos.
  - sufficient: Se enviará correcto si no existe fallo previo.
  - optional: Su respuesta solo se usa si no existe otro módulo de este servicio y tipo.

El campo control (III)

- La nueva son un conjunto de acciones a ejecutar según el valor devuelto por el módulo.
- La sintaxis es:

```
[valor1=accion1 valor2=accion2 ...]
```

- Donde valorX son unos valores predefinidos que pueden devolver los módulos y accionX es un entero positivo o una palabra clave.

El campo control (IV)

- Los posibles valores de valorX son:

- default implica valores de retorno no definidos.

El campo control (V)

- El campo accionX:
  - Si es un entero positivo N, indica que los N siguientes módulos de este tipo sean ignorados y no ejecutados.
  - Si es una palabra clave ejecuta su acción:

Valor	Descripción
ignore	El valor de retorno de este módulo es ignorado en el calculo del valor de retorno de PAM
ok	Si este módulo es correcto, PAM devolverá correcto, pero antes ejecutará el resto de módulos de este tipo.
done	Si el módulo es correcto PAM devolverá correcto inmediatamente.
bad	Si este módulo falla, PAM devolverá fallo, pero antes ejecutará el resto de módulos de este tipo.
die	Si el módulo falla PAM devolverá fallo inmediatamente.
reset	Elimina el estado de la respuesta actual y comienza su calculo a partir del siguiente módulo.

El campo control (VI)

- La equivalencia entre la sintaxis simple y la nueva es:

Sintaxis simple	Sintaxis nueva
`requisite`	`[success=ok new_authtok_reqd=ok ignore=ignore default=die]`
`required`	`[success=ok new_authtok_reqd=ok ignore=ignore default=bad]`
`sufficient`	`[success=done new_authtok_reqd=done default=ignore]`
`optional`	`[success=ok new_authtok_reqd=ok default=ignore]`

El campo camino al módulo

- Indica la localización del módulo PAM que debe ser ejecutado.
  - Si el camino empieza con / se supone un camino absoluto.
  - Si el camino no empieza con / se antepone el camino por defecto a las librerías PAM (/lib/security).

El campo argumentos del módulo (I)

- Son los argumentos que se pasan a cada módulo de PAM.
- Los argumentos son específicos de cada módulo.
- Si un módulo no reconoce un argumento lo ignora.
- Si un argumento requiere espacios entre sus componentes debe encerrarse entre [y].

El campo argumentos del módulo (II)

- Existen un conjunto de argumentos generales que reconoce todo módulo, y que son:

Argumento	Descripción
debug	Escribe en el log del sistema información para la depuración.
no_warn	No enviar mensajes de aviso.
use_first_pass	Utilizar el password obtenido previamente en lugar de solicitarlo otra vez. Si no se ha solicitado el password previamente el usuario no es autenticado.
try_first_pass	Utilizar el password obtenido previamente si este existe, o solicitarlo en caso contrario.
use_mapped_pass	No usado actualmente por ningún modulo.
expose_account	Minimizar la información enviada para solicitar la autenticación.

Módulos de la librería PAM (I)

- Existen un gran número de módulos de la librería PAM, situados por defecto dentro de /lib/security.
- No confundir con la localización de la configuración de acceso de los servicios /etc/pam.d.

Módulos de la librería PAM (II)

- pam_cracklib:
  - Tipo password.
  - Permite modificar la contraseña de un usuario comprobando que no es débil.
- pam_deny:
  - Tipos auth, account, password y session.
  - Devuelve siempre incorrecto.
- pam_env:
  - Tipo auth.
  - Asigna y/o modifica las variables de ambiente.

Módulos de la librería PAM (III)

- pam_limits:
  - Tipo session.
  - Establece el límite de recursos de un usuario en una sesión.
- pam_listfile:
  - Tipo auth.
  - Permite, mediante los datos de un fichero, admitir o denegar el acceso a determinados servicios.
- pam_loginuid:
  - Tipo session.
  - Asigna los atributos de UID al proceso que es autenticado.

Módulos de la librería PAM (IV)

- pam_nologin:
  - Tipos auth y account.
  - Comprueba si existe /etc/nologin y deja acceder solo a root.
- pam_permit:
  - Tipos auth, account, password y session.
  - Devuelve el valor correcto.
- pam_shells:
  - Tipo auth.
  - Comprueba si la shell del usuario esta en /etc/shells.

Módulos de la librería PAM (V)

- pam_succeed_if:
  - Tipo auth.
  - Comprueba las características de un usuario antes de permitir su acceso.
- pam_unix:
  - Tipos auth, account, password y session.
  - Proporciona el método clásico de autenticación basado en usuario/contraseña.

Ejemplos de ficheros PAM

- En el campo control, la palabra “include” permite incluir un fichero de configuración dentro de otro.
- No es una palabra de control, pues no indica ninguna acción.
- Se incluyen solo las líneas que correspondan al tipo indicado.
- En los ejemplos que veremos se incluye siempre el fichero system-auth.

Fichero password-auth (I)

```
#%PAM-1.0
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        required      pam_deny.so
account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     required      pam_permit.so
password    requisite     pam_cracklib.so try_first_pass retry=3 authtok_type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass
   use_authtok
password    required      pam_deny.so
session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_suceed_if.so service in crond quiet
   use_id
session     required      pam_unix.so
```

Fichero password-auth (II)

- Requerimos poder asignar valores a las variables de ambiente.

```
auth        required      pam_env.so
```

- Es suficiente con indicar de forma correcta el usuario y la contraseña, permitiendo contraseñas nulas y reutilizar los valores si ya se han solicitado.

```
auth        sufficient    pam_unix.so nullok try_first_pass
```

- Es un requisito no escribir en el log si el UID es mayor que 1000.

```
auth        requisite     pam_succeed_if.so uid >= 1000
   quiet_sucess
```

- Es un requisito denegar el acceso.

```
auth        required      pam_deny.so
```

Fichero password-auth (III)

- Requerimos que la cuenta no haya caducado.

```
account     required      pam_unix.so
```

- Es suficiente que el usuario exista en el fichero de contraseñas.

```
account     sufficient      pam_localuser.so
```

- Es suficiente no escribir nada en el log si el UID es menor de 1000.

```
account     sufficient    pam_succeed_if.so uid < 1000
  quiet
```

- Es requerido permitir el acceso.

```
account     required      pam_permit.so
```

Fichero password-auth (IV)

- Es requisito que si la contraseña ha caducado el usuario proporcione una nueva, reutilizando el usuario y teniendo tres intentos.

```
password    requisite     pam_cracklib.so
   try_first_pass retry=3 authtok_type=
```

- Es suficiente cifrar la contraseña con MD5 en shadow y permitiendo contraseñas nulas.

```
password    sufficient    pam_unix.so sha512 shadow
   nullok try_first_pass use_authtok
```

- Es un requisito denegar el acceso.

```
password    required      pam_deny.so
```

Fichero password-auth (V)

- Creamos una nueva clave de sesión.

```
session     optional      pam_keyinit.so revoke
```

- Es requisito establecer los limites de sesión del usuario.

```
session     required      pam_limits.so
```

- Comprobamos las caracteristicas del usuario.

```
session     [success=1 default=ignore]
  pam_succeed_if.so service in crond quiet use_id
```

- Es requerido ejecutar las acciones estándar de inicio de sesión en UNIX.

```
session     required      pam_unix.so
```

Servicio de SSH (fichero /etc/pam.d/sshd)

```
#%PAM-1.0
auth       required     pam_sepermit.so
auth       include      password-auth
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
session    required     pam_selinux.so close
session    required     pam_loginuid.so
session    required     pam_selinux.so open env_params
session    optional     pam_keyinit.so force revoke
session    include      password-auth
session    include      postlogin
```

- En el tipo account se requiere que se compruebe que los usuarios que no son root pueden acceder.
- En el tipo session se requiere que se asigne el UID del proceso.

Servicio de FTP (fichero /etc/pam.d/vsftpd)

```
#%PAM-1.0
session    optional     pam_keyinit.so force revoke
auth       required     pam_listfile.so item=user sense=deny
   file=/etc/vsftpd/ftpusers onerr=succeed
auth       required     pam_shells.so
auth       include      password-auth
account    include      password-auth
session    required     pam_loginuid.so
session    include      password-auth
```

En el tipo auth se requiere que se compruebe que el usuario no esta en el fichero /etc/vsftpd/ftpusers, devolviendo error si se encuentra en el mismo.

- En el tipo auth se requiere que se mire que la shell del usuario esta autorizada.
- En el tipo session se requiere que se asigne el UID del proceso.