TEMA 5: Securizando las cuentas de usuario

Modulos PAM

Pluggable Authentication Modules, son unos módulos de autenticación configurables, lo que significa que se puede cambiar la autenticación del sistema cambiando la configuración del PAM. Los módulos PAM están en /etc/pan.d. Existe un archivo PAM.conf, pero ya no se usa al estar obsoleto. Cada archivo dentro de estar carpeta configura un servicio diferente. Cada módulo tiene una serie de facilidades para cada una de las tareas con las que puede tratar PAM:

- authentication management (auth): Determina si el usuario es quien dice ser
- account management: Maneja las incidencias de disponibilidad de cuenta no relacionadas con la autenticación
- Session management: Realiza tarea asociadas con la configuración de una sesión y otras tareas como contabilización de login, establecer límites de recursos, etc…
- password management: Para cambiar el token de autenticación asociado con una cuenta.

Para cada facilidad un módulo puede ser:

- Sufficient: Si este módulo da acceso, el acceso queda garantizado, no es necesario revisar más módulos
- Requisite: Si este módulo deniega el acceso, el acceso queda denegado y no es necesario revisar más módulos
- Required: Este módulo debe garantizar el acceso y la evaluación continúa con los siguientes modilos
- Opcional: El resultado de este módulo solo será usado si el de otros módulos no es determinista.
- [new Syntax]
    - Success
    - errores varios

En el archivo /etc/pam.d/login (config para el login via terminal) se introduce lo siguiente:

```
auth requisite pam_securetty.so #Esto permite logins de root si el usuario se mete en una TTY segura
auth sufficient pam_securetty.so #Esto permite acceder solo con estar conectado desde una terminal segura.
auth requisite pam_shells.so #Si el usuario no está logueado, se rechaza.
```

En el archivo /etc/pam.de/lightdm (config para el login via interfaz gráfica) se introduce lo siguiente:

```
auth requisite pam_shells.so #Si un usuario no está en el archivo shells no podrá loguear.
```

Vulnerabilidades

- un usuario puede no ser quien dice ser:

    - Se debe endurecer la autenticación con reglas más estrictas sobre contraseñas
    - Limitar las horas a las que un usuario root puede hacer login
    - revisar los logs para detectar comportamientos anómalos
- Un usuario puede abusar de sus privilegios
    - Se deben imponer límites sobre los recursos usados por un usuario.
        - Con pam_limits se puede checkear /etc/limits que afecta a los límites
        - Poner cuotas
        - Poner límites a las aplicaciones.
- Un usuario tiene permisos que no debería tener

Securizando la autentización

Los métodos más comunes son:

- Contraseñas
    - Poner permisos más restrictivos en el archivo /etc/shadow (el que almacena las contraseñas)
        - No se almacena la contraseña cifrada, si no, la forma cifrada de un texto que usa la contraseña como clave
        - Se añade SALT
        - Se usan algoritmos deliberadamente lentos.
    - Sitios importantes
        - /etc/pam.d/login
        - /etc/pam.d/common-auth
        - /etc/pam.d/common-pasword
        - /etc/pam.d/lightdm
    - Módulos pam usados:
        - pam_unix: Definición del hash de la contraseña y las características de la contraseña
        - pam_pwquality: establece características de las contraseñas
        - pam_pwhistory: deshabilita el reciclado de contraseñas
        - pam_securetty: Limita los logins a root desde ciertos dispositivos
        - pam_faildelay: Establece una demora entre fallos de logueo
        - pam_ google_authenticator: verificación en dos pasos de google
- certificados digitales
- Datos biométricos
- Clave física

Verificación en 2 Pasos

Configuración de la verificación en 2 pasos:

```
google-authenticator
#Pregunta si queremos los tokens basados en tiempo
y #Le damos
#Sale un QR que podemos escanear con el authenticator de google.
#Se introduce el código que nos da el authenticator.
```

Restricted Shells

El usuario no puede usar ni modificar variables de entorno a menos que se las permitamos una por una. Generalmente se le asigna al usuario una ubicación con los símbolos lógicos de lo que se le permite usar.

Root

Se debe deshabilitar el login directo al root de la máquina, de forma que para usar el root tendríamos que usar “su - root” o “sudo” desde otro usuario.

El comando Sudo

Con sudo ejecutas con otro usuario, aunque en sistemas como ubuntu, sudo ejecuta siempre como root, mientras que lo normal es que sudo deje ejecutar con un usuario determinado. Sudo se rige por /etc/sudoers

```
usuario abyecti=(root) shutdown #Quien, máquina = como quien y que comando
tiene  permitido, en este caso esto significa que el usuario antonio como el
usuario abyecto puede ejecutar como root el shutdown.
usuario abyecti=(root) ALL #Permite hacer todo como root al usuario usuario.

#Dar permisos de apagado a un grupo:
User_Alias DOWNDOERS = pepe, pepa, user2 #Se crea el grupo
Cmnd_Alias POWERDOWN = /sbin/shutdown, /sbin/halt, /sbin/reboot,
/sbin/restart #Grupo de permisos
DOWNDOWERS rutercillo = (root) POWerDOWN #Se asigna el grupo de permisos al
grupo de usuarios en la máquina rutercillo para que los puedan ejecutar como
root.
```

From:
https://knoppia.net/ - **Knoppia**

Permanent link:
https://knoppia.net/doku.php?id=master_cs:fortificacion:tm5&rev=1740416635

Last update: **2025/02/24 17:03**