**Administración de Sistemas Operativos - 1a Evaluación (RA 4 – CE d, e, f)**

**Unidad Didáctica 4. Configuración multiusuario centralizada**

**Ejercicio 1. Investiga sobre qué es PAM.**

Privileged Access Management (PAM) es una solución de seguridad de identidad que ayuda a proteger las organizaciones contra las ciberamenazas al supervisar, detectar y evitar el acceso con privilegios no autorizados a recursos críticos. PAM funciona mediante una combinación de personas, procesos y tecnologías que ofrece visibilidad sobre quién está utilizando cuentas con privilegios y qué están haciendo mientras están conectados. Limitar el número de usuarios que tienen acceso a funciones administrativas incrementa la seguridad del sistema, mientras que las capas adicionales de protección mitigan la filtración de información llevada a cabo por actores de amenazas.

#### **¿Qué es el acceso con privilegios?**

Se refiere a cuentas con capacidades avanzadas, como la administración de sistemas, que pueden agregar, modificar o eliminar usuarios y acceder a áreas restringidas. Ejemplo: los administradores en Windows o usuarios raíz en Linux.

#### 

#### **Componentes de PAM:**

1. **Administración de cuentas privilegiadas:**
   * Gestión segura de credenciales (contraseñas) en bóvedas protegidas.
   * Aplicación de políticas como autenticación multifactor y rotación automática de contraseñas.
2. **Administración de sesiones privilegiadas:**
   * Monitoreo y auditoría de actividades en cuentas privilegiadas.
   * Alertas ante actividades anómalas.
   * Grabación de sesiones para investigación y cumplimiento normativo.
3. **Ciclo de vida de cuentas:**
   * Automatización en creación, modificación y eliminación de cuentas privilegiadas

Es crucial porque reduce riesgos de filtraciones de datos, limita accesos no autorizados, disminuye la superficie de ataque y asegura el cumplimiento normativo. PAM protege las identidades críticas en entornos tecnológicos, minimizando amenazas internas y externas.

**Ejercicio 2. Busca información sobre los tipos de módulos y las banderas de control utilizadas en PAM.**

**TIPO:**

**Módulos de autenticación; auth:**

Autenticar usuarios y establecer, renovar o destruir credenciales. Estos módulos identifican al usuario basándose en su autenticación y credenciales.

* *pam\_sm\_authenticate*
* *pam\_sm\_setcred*

**Módulos de gestión de cuentas; account:**

Determinar la validez de la cuenta de usuario y el acceso posterior después de la identificación del módulo de autenticación. Las comprobaciones realizadas por estos módulos suelen incluir restricciones de caducidad de cuenta y contraseña.

* *pam\_sm\_acct\_mgmt*

**Módulos de gestión de sesiones; session:**

Iniciar y terminar sesiones de usuario. Además, se puede proporcionar soporte para la auditoría de sesión.

* *pam\_sm\_open\_session*
* *pam\_sm\_close\_session*

**Módulos de gestión de contraseñas; password:**

Realice la modificación de contraseña y la gestión de atributos relacionados.

* *pam\_sm\_chauthtok*

**BANDERAS DE CONTROL:**

En PAM, las banderas de control en /etc/pam.conf definen si los módulos siguientes se ejecutan (comportamiento de continuación) y cómo se manejan los errores (comportamiento de fallo). Un fallo requerido provoca el fallo de la solicitud, mientras que uno opcional no siempre lo hace. Estas banderas son clave para controlar el flujo de autenticación

**required:**

Los módulos con este indicador deben procesarse correctamente. Si fallan, el resto de módulos required se procesan antes de informar al usuario sobre el error.

**requisite:**

Similar a required, pero si un módulo falla, se notifica de inmediato y no se ejecutan más módulos. Útil como filtro para las condiciones necesarias.

**sufficient**

Si se ejecuta correctamente, la autenticación se considera exitosa y no se procesan más módulos (a menos que antes haya fallado un módulo required). Si falla, no interrumpe el proceso, y se siguen ejecutando los módulos restantes.

**optional**

Su éxito o fallo no afecta al resultado general. Ideal para módulos informativos, como mostrar mensajes al usuario.

**include**

Permite insertar la configuración de otro archivo especificado en la ubicación actual.

**MODULE\_PATH:**

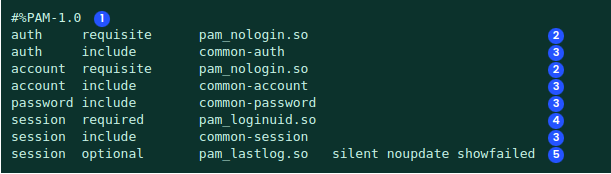
Especifica la ubicación del módulo PAM. Si el módulo está en el directorio predeterminado (/lib/security o /lib64/security), no es necesario incluir la ruta completa.

**MODULE\_ARGS:**

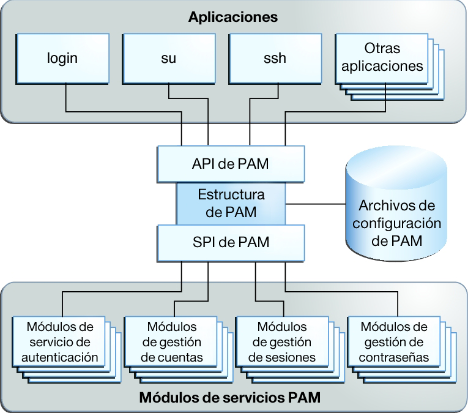
Contiene una lista de opciones separadas por espacios para influir en el comportamiento de un módulo PAM, como debug (habilita la depuración) o nullok (permite el uso de contraseñas vacías).

Los archivos de configuración globales de PAM en /etc/security centralizan los ajustes para simplificar el mantenimiento. Las configuraciones comunes permiten actualizaciones automáticas en todas las aplicaciones, y la herramienta pam-config automatiza su gestión, reduciendo la necesidad de intervención manual

*Configuración de PAM para sshd (/etc/pam.d/sshd)*



**Ejercicio 3. Explica la arquitectura asociada.**



La arquitectura funciona de la siguiente manera:

* Las aplicaciones se comunican con la estructura del PAM a través de la interfaz de programación de aplicaciones (API) del PAM.
* Los módulos de servicio del PAM se comunican con la estructura del PAM a través de la interfaz del proveedor de servicios (SPI) del PAM. Para obtener más información, consulte la página del comando man pam\_sm(3PAM).

Los administradores pueden configurar una o más series de módulos para gestionar los requisitos del sitio. Esta serie de módulos se denomina pila del PAM. La pila se evalúa en orden. Si una aplicación necesita más de una pila del PAM, el desarrollador de la aplicación debe crear más de un nombre de servicio. Por ejemplo, el daemon sshd proporciona y requiere varios nombres de servicio para el PAM. Para conocer la lista de nombres de servicio del PAM para el daemon sshd, busque la palabra PAM en la página del comando man sshd.

**Ejercicio 4. Expón qué directorios y ficheros son de gran importancia.**

Las aplicaciones del sistema, como login y ssh, que utilizan la estructura del PAM están configuradas en los archivos de configuración del PAM en el directorio /etc/pam.d. El archivo /etc/pam.conf también se puede utilizar. Los cambios que se realizan en estos archivos afectan a todos los usuarios del sistema.

Además, el directorio **/etc/security/pam\_policy** contiene archivos de configuración del PAM. Estos archivos abarcan varios servicios y están diseñados para la asignación por usuario. Los archivos de este directorio no deben modificarse.

* **Directorio /etc/pam.d**: Este directorio contiene los archivos de configuración específicos de cada servicio que utiliza PAM. Cada archivo define la pila de módulos para autenticar, autorizar, gestionar contraseñas y sesiones de ese servicio en particular.
* **Archivo /etc/pam.conf:** Este archivo es una alternativa al directorio /etc/pam.d/ y es utilizado principalmente en implementaciones antiguas de PAM. En sistemas modernos, suele estar vacío o sin uso.
* **Directorio /etc/security/:** Este directorio incluye archivos auxiliares que trabajan junto con los módulos PAM para definir políticas específicas de seguridad y personalización.
  + *limits.conf:* Define restricciones de recursos por usuario o grupo
  + *time.conf:* Controla el acceso según horarios definidos
  + *access.conf:* Controla el acceso según usuarios o grupos.
* **Archivo /etc/login.defs:** Define políticas relacionadas con el manejo de contraseñas y cuentas de usuario, como longitudes mínimas de contraseña, duración máxima de las contraseñas, UID mínimos, etc.
* **Directorio /lib/security/ o /usr/lib/security/:** Este directorio contiene los módulos dinámicos de PAM, los cuales tienen la extensión .so. Estos módulos implementan diferentes funcionalidades, como autenticación local, autenticación remota etc…
* *pam\_unix.so:* Maneja autenticación mediante contraseñas locales en /etc/passwd y /etc/shadow.
* *pam\_tally2.so:* Realiza un seguimiento de intentos fallidos de inicio de sesión.
* *pam\_ldap.so:* Proporciona autenticación basada en un servidor LDAP.
* **Archivo /var/log/secure o /var/log/auth.log:** Contienen registros de eventos relacionados con la autenticación, como intentos fallidos, escaladas de privilegios, etc.

**Ejercicio 5. Crea una tabla que contenga los 10 módulos de PAM que más te hayan**

**llamado la atención. Cada fila deberá contener el nombre del módulo, descripción,**

**opciones y varios ejemplos de uso.**

| **Módulo** | **Descripción** | **Opciones** | **Ejemplos de uso** |
| --- | --- | --- | --- |
| *pam\_unix* | Autentica usuarios con contraseñas locales. | nullok, md5, shadow, debug. | Configurar autenticación básica en sistemas Linux con contraseñas locales. |
| *pam\_deny* | Deniega el acceso siempre, usado para bloquear rutas de autenticación. |  | Utilizar como último módulo en la pila para garantizar que cualquier acceso no gestionado sea denegado. |
| *pam\_faildelay* | Configura un retraso en las respuestas de fallos de autenticación. | debug, delay=n (n en microsegundos). | Añadir un retraso de 2 segundos (delay=2000000) para dificultar ataques de fuerza bruta. |
| *pan\_env* | Configurar variables de entorno. | readenv, debug, user\_readenv. | Establecer variables como PATH o HOME al iniciar sesión, definiéndose en pam\_env.conf. |
| *pam\_tally2* | Lleva un registro de intentos fallidos de inicio de sesión. | deny, unlock\_time, magic\_root, debug. | Bloquear usuarios después de 5 intentos fallidos, desbloqueándolos tras 10 minutos. |
| *pam\_limits* | Aplica límites a los recursos del sistema para los usuarios. | conf=/ruta/archivo, debug. | Configurar límites en el uso de memoria, CPU o procesos concurrentes en /etc/security/limits.conf. |
| *pam\_motd* | Muestra mensajes al usuario al iniciar sesión. | noupdate, debug. | Mostrar un mensaje de bienvenida definido en /etc/motd o un mensaje personalizado para todos los usuarios. |
| *pam\_time* | Controla el acceso según la hora o el día. | conf=/ruta/archivo, debug. | Permitir acceso solo durante horarios laborales, configurado en /etc/security/time.conf. |
| *pam\_sss* | Integra autenticación con servicios de terceros, como LDAP o SSSD. | min\_uid, debug, try\_cert\_auth. | Autenticar usuarios utilizando un servidor LDAP corporativo. |
| *pam\_exec* | Permite ejecutar comandos o scripts durante el proceso de autenticación. | seteuid, debug. | Ejecutar un script para enviar notificaciones al administrador al inicio de sesión de un usuario crítico. |

**Ejercicio 6. Realiza las siguientes configuraciones:**

* **Deshabilita el acceso vía SSH de cualquier usuario que no sea el usuario root. El sistema deberá de mostrar un mensaje cuando un usuario intente acceder.**

Comenzaremos editando el archivo /etc/pam.d/sshd para añadir o editar la línea:



Una vez aplicada esta línea, reiniciamos el servicio con: *sudo systemctl restart sshd*

* **Cambia el mensaje del día cuando un usuario se conecte vía SSH.**
* **Configura franjas horarias en las que los usuarios pueden conectarse al sistema.**
* **Define alguna variable de entorno que el usuario se encontrará al realizar la conexión.**

**Ejercicio 7. Visita las siguientes páginas:**

* <https://github.com/mlabouardy/pam-qrcode>
* <https://github.com/nahil1/pam-bluetooth>

**Además de los anteriores, busca módulos PAM que te resulten interesantes. Explica qué te ha llamado la atención cada uno de ellos.**

* <https://github.com/google/google-authenticator-libpam>

Permite implementar la autenticación de dos factores en sistemas Linux mediante códigos generados por aplicaciones como Google Authenticator o compatibles con TOTP. Añade una capa extra de seguridad protegiéndote contra accesos no autorizados, incluso si la contraseña principal se compromete. La configuración es sencilla gracias a la generación de códigos QR que facilitan el emparejamiento inicial. Respaldado por Google, ofrece confiabilidad y actualizaciones regulares

* <https://github.com/muesli/pam-beacon>

Utiliza dispositivos Bluetooth Low Energy, como teléfonos móviles o pulseras inteligentes, para la autenticación automática en sistemas Linux. Esta tecnología permite un acceso rápido y manos libres sin necesidad de ingresar contraseñas, mejorando la seguridad al asegurar el acceso solo si el dispositivo autorizado está presente. Integra tecnologías modernas como BLE, proporcionando una alternativa innovadora a los métodos tradicionales de autenticación