

TALLER DE SEGURIDAD INFORMÁTICA

CURSO 2020

Informe - Laboratorio 02

Grupo 03

Autores:

Agustín RIEPPI

4.754.760-1

Guillermo COELHO

5.043.831-6

27 de septiembre de 2020

Índice

1. Introducción	2
2. Implementación	2
2.1. Hipótesis de trabajo	2
2.2. Escenario	3
2.3. Modulo	3
2.3.1. Interfaz autenticación	3
2.3.2. Interfaz password	4
2.4. Configuración del stack	5
2.5. Programa de testeo	5
2.6. Integración al comando passwd	6
3. Mejoras y trabajo a futuro	6
4. Conclusión	6

1. Introducción

Este laboratorio tiene como fin entender como es el correcto funcionamiento y la implementación de algunas de las características mas comunes en los módulos PAM (Pluggable Authentication Modules) [1].

La idea principal de este framework es poder generar módulos totalmente independientes y que puedan ser usados en cualquier contexto, siempre que se cumplan ciertas pre-condiciones. Cada uno de estos modulo pueden proveer 1 una mas interfaz, estas pueden ser autenticación, account, session y password. En en el siguiente libro[] se encuentra una descripción mas extensa de que significa cada una de estas interfaces y como se pueden combinar los módulos.

2. Implementación

Para este laboratorio en particular se desarrolla un modulo que implementa la interfaz de password y autenticación. El objetivo de este nuevo modulo es que sea flexible para poder ser utilizado con cualquier base de datos que contenga las contraseñas. Esta base de datos debe cumplir la siguiente estructura:

user:{SCHEME}pass:::allow_hosts=192.168.0.1,etc:allow_group=group1,etc

Por lo cual cuando un usuario se quiere autenticar no solo debe ingresar las correctas credenciales sino que además debe estar dentro de los grupos y hosts habilitados. También a la hora de cambiar la contraseña se le pide al usuario que ingrese el método de cifrado para aplicar a la contraseña, puede ser (MD5, SHA215, Plain) además el administrador puede cambiar el esquema de cifrado a LK, lo cual significa que el usuario esta bloqueado.

2.1. Hipótesis de trabajo

A las hipótesis definidas por el los docentes se agregan las siguientes:

- El grupo utilizado para validar que pertenece a los grupo habilitados es el grupo real que ejecuto el proceso.

- Se valoran las ventajas, como la flexibilidad, que pueden llegar a tener los grupos configurados por el administrador pero en este caso no era el objetivo configurarlos, por lo que se utilizaron los de sistema.
- La aplicación permitirá cambiar la contraseña a un usuario distinto al autenticado anteriormente.

2.2. Escenario

El escenario de elección fue trabajar sobre una maquina física con una maquina virtual corriendo Linux con una arquitectura de 64 bits. Las distribuciones utilizadas para la verificación son Ubuntu 20.04.1 LTS y Fedora 32, tomando en cuenta que soportan las funcionalidades necesarias.

2.3. Modulo

El modulo que se implementa tiene 2 interfaces a continuación se detalla a grandes rasgos la implementación realizada. Ambas lo primero que chequean son los argumentos que le pasaron al invocarlos. Como es un requerimiento un de los argumento debe ser un string que contenga "filedb.^{el} cual debe venir seguido de la ruta absoluta hacia la ubicación de la base de datos. El otro argumento que se puede recibir es el string "debug", cuando se le pasa este argumento el modulo cada vez que el resultado no sea exitoso va a reportar el incidente en log del sistema informando detalladamente que fue lo que fallo.

En ambos casos el manejo de la base de datos se realiza bloqueando el archivo antes de leerlo y/o escribirlo, para evitar problemas de concurrencia.

2.3.1. Interfaz autenticación

En esta interfaz lo primero que se realiza es pedir el usuario y la contraseña. Para pedir el usuario se hace uso del método proporcionado por el framework PAM llamado `pam_get_user()`, el cual le solicita al proceso que invoco el modulo que ingrese

el username, lo mismo se realiza con la contraseña utilizando el método `pam_get_authtok`.

Luego de obtener el username va a buscar a la base de datos toda la información asociada a este. Con la información obtenida se procede a validar la contraseña, los grupos habilitados y los host habilitados.

Validar la contraseña: Para validar la contraseña se chequea que tipo de esquema tiene, si utiliza alguno que sea de hash, se procede a aplicar el hash, y se compara con la contraseña almacenada, lo mismo se realiza si el esquema es texto plano pero sin aplicar el hash. En el caso de que el esquema sea LK se procede a denegar el acceso

Validar la hosts: Para validar los hosts, se consiguen todas las IPv4 de la maquina que esta ejecutando el proceso, luego se verifica si alguna esta dentro de los hosts habilitados.

Validar la grupos: Para validar los grupos, se consigue el id real del grupo del usuario que ejecuto el proceso, luego se verifica que este en los grupos habilitados.

Si todas las validaciones son positivas se procede a retornar que el modulo finalizo con éxito, en caso contrario se retorna el error que corresponda.

2.3.2. Interfaz password

Esta interfaz es llamada 2 veces, una para validar las antiguas credenciales y otra para actualizar las credenciales.

La primera llamada es análoga a la interfaz de autenticación exceptuando las validaciones de host y grupo.

La segunda llamada se pide ingresar la nueva contraseña mediante el método `pam_get_authtok` pasándole al mismo la flag `PAM_AUTHTOK` para así pedir que se ingrese 2 veces la nueva contraseña y así chequear que sean iguales.

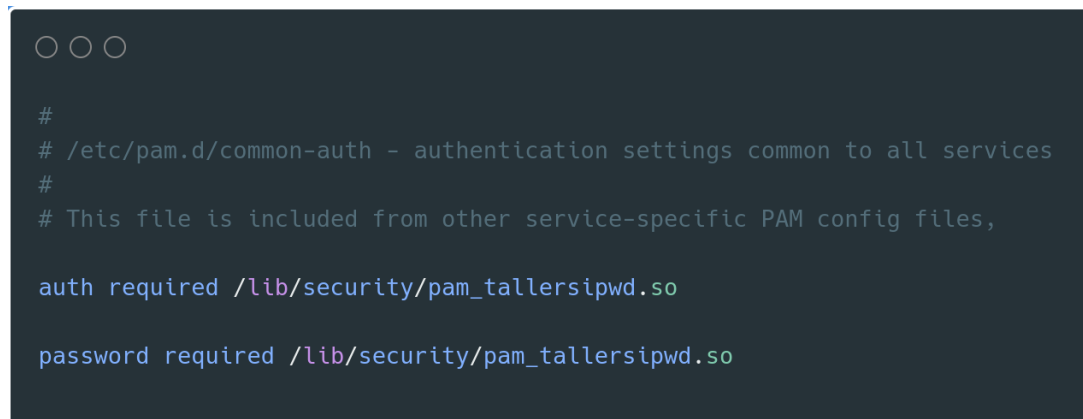
Luego se pide el esquema de encriptado que se desea para almacenar la nueva contraseña, para esto se emplea la función de converse proporcionada por PAM.

Si el usuario elige un esquema de texto plano se valida que no tenga caracteres que puedan romper la base de datos, estos son (`{,:`) Teniendo entonces la nueva contraseña y el esquema se hashea la nueva contraseña en el caso que corresponda, codificándola en hexadecimal. Finalmente la base de datos es actualizada con los nuevos datos del usuario correspondiente.

2.4. Configuración del stack

En este caso el archivo de configuración del modulo que se va a implementar se va a ajustar a las interfaces que se hayan desarrollado, como se menciona anteriormente, authentication y password.

En ambos casos se usa la flag required para que funcione solo en que caso que se hayan completado sin errores.



```
○ ○ ○  
  
#  
# /etc/pam.d/common-auth - authentication settings common to all services  
#  
# This file is included from other service-specific PAM config files,  
  
auth required /lib/security/pam_tallersipwd.so  
  
password required /lib/security/pam_tallersipwd.so
```

Figura 1: Archivo de configuración del modulo

2.5. Programa de testeo

Se desarrolla un programa para validar el correcto funcionamiento del modulo. Este programa invoca el archivo de configuración que se mostró en la sección anterior. Algunas de las pruebas realizadas para cada interfaz fueron: **Autenticación:**

- Usuario y contraseña correctos, grupo y host validos ->Éxito
- Usuario correcto y contraseña incorrecta ->Error
- Usuario vacío ->Error
- Usuario correcto y contraseña incorrecta ->Error
- Usuario y contraseña correctos, grupo valido y host invalido ->Error
- Usuario correcto y contraseña correctos, host valido y grupo invalido->Error

- Usuario correcto y contraseña correctos, con esquema LK ->Error

Password: Estas se realizan siempre con el caso de éxito Usuario y contraseña correctos, para así validar el ingreso de nueva contraseña

- Nueva contraseña correcta y esquema correcto ->Éxito
- Nueva contraseña correcta y esquema incorrecto ->Falla
- Nueva contraseña que no coincida entre las 2 entradas ->Falla
- Nueva contraseña con caracteres ilegales ([:,) y esquema texto plano ->Falla

2.6. Integración al comando passwd

El modulo implementado puede sustituir la interfaz del modulo pam_unix en la configuración que usa el comando passwd. Aunque la validación y modificación de credenciales se haga en la base de datos que se le pasa como argumento, el comando passwd solo aceptar username que se encuentre en el archivo /etc/passwd

3. Mejoras y trabajo a futuro

Mejoras hay una cantidad muy grande pero se detallan alguna que resultan de mayor importancia

- Mejor manejo a la hora de actualizar la base de datos
- Mejorar la información brindada usando la flag debug
- Mejor manejo de memoria y mas cuidado a la hora de cortar al ejecución, para que no quede información "libre"

4. Conclusión

En este laboratorio se puede apreciar que si bien un modulo puede ser implementado sin grandes dificultades, el mayor problema radica en respetar los requerimientos de cada interfaz proporcionada. Además por mas simple que sea un modulo

hay que hacer un esfuerzo muy importante para cuidar todos los aspectos de seguridad dado que sobre estos módulos cae la responsabilidad de cuestiones de seguridad del sistema operativo.

Por otro lado desarrollar programas que hagan uso de módulos existentes resulta muy fácil y seguro, quedando a al vista los beneficios de los módulos PAM.

Referencias

- [1] Pluggable Authentication Modules for Linux - Man Page,
<https://linux.die.net/man/8/pam> Visitada setiembre de 2020.
- [2] Gnupg - The Libgcrypt Reference Manual:
<https://gnupg.org/documentation/manuals/gcrypt.pdf>. Visitada setiembre de 2020.