Doble factor de autenticación independiente de SO

Roselló Morell, Sergio
sergio.rosello@live.u-tad.com

June 6, 2018

Contents

1 Agradecimientos

Debo agradecer a Eduardo Ariols, mi tutor del trabajo todo el apoyo y consejos dados. Estoy seguro de que sin su ayuda, este trabajo no hubiese llegado a su nivel actual. Durante el proceso de elección del trabajo, me ayudó a darme cuenta de lo que quería hacer exactamente y desde ese momento, no ha parado de inspirarme con distintas formas de ver las cosas. De eso, le estoy muy agradecido. También quiero agradecer a la universidad el buen trabajo a la hora de escoger al personal docente de mi grado, puesto que en todo momento han demostrado más que profesionalidad y compañerismo hacia mi y mis compañeros de carrera.

2 Resumen

Este documento explica al lector la experiencia que he tenido durante el periodo de realización del trabajo de final de grado. Este trabajo trata sobre la autenticación de un usuario a un sistema GNU/Linux, en concreto, mediante una llave USB.
Durante la fase de investigación de las tecnologías existentes, encontré algunas que ofrecían usa solución elegante, mediante Dbus pero acabando la fase de investigación encontré un proyecto llamado Pluggable Authentication Module que redefinió la forma en la que planteaba el trabajo. Ésta es la forma por defecto de autenticar a los usuarios que tienen la mayoría de sistemas GNU/Linux.
Este trabajo, al principio con enfoque mucho más práctico ha acabado teniendo un enfoque investigativo puesto que para implementar el módulo de autenticación, he tenido que construir una base fuerte sobre la que sentirme cómodo. Esta base es la que he tenido que esforzarme a entender puesto a que sin ella, el trabajo realizado, aunque funcionalmente completo, no me hubiese sido ni la mitad de estimulante e interesante.

Abstract

This document reports my experience as I work on creating a USB-centric authentication method for GNU/Linux. During the research phase, I came across several elegant implementations, all of them worked with Dbus. During the final stages of this period, I discovered Pluggable Authentication Module which changed my whole perspective on this project. Most of GNU/Linux systems use this module to enable authentication for their users.
At the start of this project I would've expected to code a lot more, but now I realise that without a solid foundation, I may have been able to do what I had proposed, but I would not have the understanding on how the Pluggable Authentication Module fits into the whole equation and the many benefits it provides. This, I think is the point of this work.

Abreviaciones y tecnicismos

Access Control List Modelo de permisos POSIX-compliant simple pero potente. Uno de las primeras formas de implementaciones de privilegios.

dbus Desktop bus. Un sistema para habilitar la comunicación entre programas.

Firewalls Parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaicones no autorizadas.

GNU/Linux Combinacón del kernel creado por Linus Torvalds y de los programas creados por el proyecto GNU.

kerberos Protocolo de autenticación de red.

Lightweight Directory Access Protocol Protocolo de acceso y búsqueda de datos a la base de datos distribuida X.500.

Microsoft Active Directory Protocolo de acceso y búsquedade datos en la base de datos distribuida X.500.

Network Information Service Base de datos distribuida en una red, que almacena las credenciales y permisos de los usuarios de la red.

Pluggable Authentication Module Módulo que funciona como adaptador entre los programas de verificación nuevos como por USB o llave maestra y los programas que gestionan la autenticación. Si no existiera, cada vez que se crea un nuevo esquema de autenticación, se debería de actualizar todos los programas que usan ese servicio.

Role Based Access Control Sistema que trata de gestionar la seguridad de una forma basada en roles y no tan granular como ACL.

Secure SHell Programa que permite conectarse a un sistema mediante una conexión segura.

SetGid Set group ID on execution: Si un archivo ejecutable contiene este bit, permite a usuarios ejecutar el archivo con los mismos privilegios que el grupo que posee el archivo.

SetUid Set user ID on execution: Si un archivo ejecutable contiene este bit, permite a usuarios ejecutar el archivo con los mismos privilegios que el usuario que posee el archivo.

Simple Authentication and Security Layer forma de añadir autenticación y seguridad a protocolos basados en red.

StickyBit Solo permite modificar el archivo/directorio por el usuario que lo ha posee.

Telecomunication Network Protocolo de red que nos permite aceder a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. Tambien es el nombre de uno de los programas informáticos que implementan el protocolo.

USB Universal Serial Bus: Interfaz que permite la conexión de periféricos a diversos dispositivos.

X.500 Es una base de datos distribuida que ofrece la posibilidad de buscar información por nombre (páginas blancas) y buscar información (Páginas amarillas).

3 Estado del arte

En esta sección, describiré las distintas tecnologías de autenticación existentes, los problemas actuales de esas tecnologías, defenderé la necesidad de una seguridad más agresiva, tanto para los usuarios normales como para las grandes empresas y desglosaré las distintas formas de conseguir esta seguridad actualmente.

3.1 Tecnologías y protocolos

La forma en la que se autentica la identidad de los usuarios de sistemas ha ido evolucionando desde que se vio que era necesaria.

3.1.1 Inicios de la gestión de permisos

Cada objeto tiene asociada una tabla de 9 bits, los tres primeros indican los privilegios de lectura, escritura y ejecución del usuario que posee el objeto. Los tres siguientes son para la lectura, escritura y ejecución de los usuarios pertenecientes al grupo que posee el objeto y los tres últimos son de lectura, escritura y ejecución de los usuarios que no pertenecen a ninguno de las dos primeras categorías. Esta categoría se llama *others*. Además de estos 9 bits, también pueden incluir el SetUid, SetGid y el StickyBit. A pesar de ser un sistema muy simple de gestionar privilegios, cumple la mayoría de escenarios posibles en sistemas UNIX e incluso a día de hoy, se sigue usando en todos los sistemas GNU/Linux ya que proporciona una forma sencilla y eficiente de visualizar los privilegios de los objetos y modificarlos. Esta forma de gestionar los privilegios de los objetos se puede denominar Access Control List.

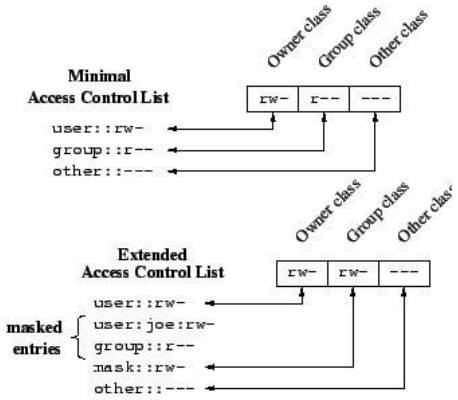

Figure 1: Access Control List

3.1.2 Role Based Access Control

Este esquema de seguridad está diseñado para organizaciones o sistemas en los que van a interactuar distintos usuarios con una gran cantidad de datos. El sistema defiende que, en lugar de tener una tabla por cada objeto, definiendo la forma que tienen los usuarios de interactuar con él, se deberían establecer una serie de transacciones, que dependiendo del rol serán distintas. Estas transacciones, una vez definidas cambian poco porque un usuario específico va a usar unos documentos específicos, dependiendo de la responsabilidad que tenga en la organización. En la imagen 2 se puede ver claramente como dependiendo del rol vas a poder acceder a ciertos objetos.

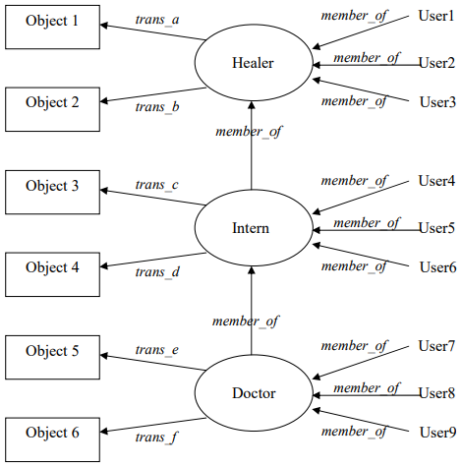

Figure 2: Role Based Access Control

Dos de las ventajas de este sistema son que cumplir el principio de de menor privilegio es relativamente sencillo, ya que se puede conseguir no proporcionándole al usuario más transacciones de las que debe tener. Otra de las ventajas, innata, de Role Based Access Control es la separación de deberes. Esto es: En el caso de tener que realizar un transferencia bancaria, nunca se debería de poder proporcionar al mismo individuo el control de todo el flujo, ya que se le está dando la oportunidad de cometer algún tipo de irregularidad. Con RBAC puedes asignar dos transacciones, una que permita a un usuario solicitar una transferencia y otra que permita a un usuario validar la transferencia.

3.1.3 Lightweight Directory Access Protocol

Este protocolo está diseñado para permitir acceso a directorios complacientes con el estándar X.500 *Directory Access Protocol* a sus usuarios. Uno de varios protocolos que tiene una aplicación para autenticarse con el fin de acceder al directorio X.500[M W97]. La forma de acceder a los datos cambia según el

protocolo de acceso a X.500. Por ejemplo, en esta implementación, la forma de acceder a los datos sería:

```
cn=Rosanna Lee, ou=People, o=Sun, c=us
```

mientras que la implementación de Microsoft sería:

```
/c=us/o=Sun/ou=People/cn=Rosanna Lee
```

Cada uno de estos protocolos define una forma de "buscar" en X.500 información. Podemos ver que la implementación de Lightweight Directory Access Protocol está ordenada de derecha a izquierda, separada por el carácter (",") mientras que la implementación de Microsoft, Microsoft Active Directory está ordenada de izquierda a derecha y separada con el carácter ("/"). Aunque no sea un método de autenticación que sucede en el mismo sistema, me parece que es lo suficientemente interesante como para incluirlo ya que es un ejemplo de autenticación en red.

3.1.4 Simple Authentication and Security Layer

Es un protocolo que proporciona métodos de añadir autenticación a protocolos de red mediante identificación y autenticación de los usuarios conectados al servidor. Además, gestiona el nivel de seguridad que se desea establecer para las futuras interacciones entre el servidor y el usuario conectado. Si se llega a la conclusión de que si que se requiere una capa de seguridad, esta se añade entre el propio protocolo y la conexión.[Mye97] Las distintas formas de autenticarse a un servidor con este protocolo son:

- **Anonymous:** Usado para autenticar a clientes a servicios anónimos. El cliente envía un token (Correo electrónico) para permanecer identificado con el servidor. Es una forma sencilla y rápida de implementar, pero no es segura.
- **CRAM-MD5:** Usa el nombre de usuario y una contraseña para autenticar a los usuarios, pero solamente se transfiere la contraseña hasheada. Esto implica que no se pueden usar métodos de autenticación normal como Pluggable Authentication Module, que no soporta extracción de contraseñas. Es una forma simple y segura de autenticarse con el servidor.
- **KERBEROS_V4:** Forma de autenticación fiable. Rápida, pero complicada de implementar. Muy segura.
- **por defecto (autenticación y autorización):** Usa el nombre de usuario y la contraseña para autenticar a los usuarios. La forma más rápida y sencilla pero poco segura.
- **SCRAM-MD5:** Deprecada.

- **DIGEST-MD5:** Basada en CRAM-MD5 pero da soporte a más características. Solo se transfieren las contraseñas hasheadas, por tanto no se puede usar Pluggable Authentication Module como backend. Es simple y seguro.
- **LOGIN:** Usa nombre de usuario y contraseña para autenticar a los usuarios. Rápida, simple de implementar pero nada segura.
- **OTP:** One Time Password
- **SECURID:** Usa una clave de un dispositivo hardware para autenticar a los usuarios. Buena velocidad, difícil de implementar pero buena seguridad.

Este protocolo ofrece una ventaja muy significativa. Proporciona a los desarrolladores la posibilidad de implementar su propio mecanismo para que utilice SASL.

3.1.5 Kerberos

Kerberos es un protocolo de autenticación de red diseñado para proporcionar un alto nivel de seguridad en la forma en la que el cliente y el servidor se autentican. Para conseguir esto, usa criptografía de llave secreta.[MIT98]
El protocolo parte con la base de que internet no es un sitio seguro. Muchos protocolos ni siquiera usan algún tipo de seguridad. Existen herramientas con intención maliciosa para sacar las contraseñas de la red, por tanto, transferir credenciales descifradas a través de la red, es una práctica muy insegura. Algunas páginas usan Firewalls para solucionar sus problemas de seguridad, pero los Firewalls suponen que el problema está en el exterior cuando deberían asumir que se pueden vulnerar desde dentro también. Los firewalls, además los inconvenientes de los firewalls son inaceptables porque restringen la forma que tienen los usuarios de acceder a internet.
Kerberos es la solución para este tipo de problemas de seguridad. El protocolo de kerberos usa una criptografía muy fuerte para que tanto el servidor como el cliente puedan comunicarse a través de una red insegura. Tras haberse autenticado mediante Kerberos pueden cifrar la comunicación entre ellos.

3.1.6 Network Information Service

Proporciona información a los sistemas que tiene que tener para que cualquier usuario pueda autenticarse en cualquier sistema de esa misma red. Por tanto, debe mantener sincronizados y actualizados datos como:

- Nombres de usuario, contraseñas y directorios principales de cada usuario (/etc/passwd)
- Información de grupos (/etc/group)
- Nombres de sistemas y direcciones IP (/etc/hosts)

Esta información la administra el servidor central. Dependiendo del tamaño de la red, el administrador de sistemas puede decidir replicarla en más ordenadores (esclavos) que se mantienen actualizados siempre con el servidor maestro (Cada vez que este se actualiza) Una de las ventajas de esto es que al estar los datos distribuidos, si se cae el servidor maestro, los usuarios de la red no sufren ningún percance, ya que la información está reflejada en los esclavos. Otra de las ventajas de mantener la información distribuida es que los esclavos también pueden responder a peticiones de clientes, por tanto, si un esclavo tarda menos en responder que el servidor, este puede facilitar la información al cliente.
La diferencia entre NIS y NIS+ es que el último implementa muchas mejoras, incluida entre ellas, la posibilidad de tener dominios jerárquicos.
Sin embargo, la mayor parte de administradores de sistemas recomendarían usar NIS, ya que es bastante más sencillo de administrar.

3.1.7 Secure SHell

Una de las formas que han tenido los usuarios de conectarse de forma remota con un sistema ha sido el Telecomunication Network. Esta opción, aunque válida, es poco segura porque la comunicación entre la máquina y el usuario se envía sin cifrar a través de la red.
Se desarrolló SSH para evitar esto.

3.1.8 Pluggable Authentication Module

Bibliografía

[M W97] S. Kille M. Wahl T. Howes. *Lightweight Directoyr Accesss Protocol (v3)*. Dec. 1997. URL: https://tools.ietf.org/html/rfc2251.

[Mye97] J. Myers. *Simple Authentication and Security Layer (SASL)*. Oct. 1997. URL: https://tools.ietf.org/html/rfc2222.

[MIT98] MIT. *Kerberos, the network authentication protocol*. July 1998. URL: https://web.mit.edu/kerberos/.