Doble factor de autenticación independiente de SO

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

June 5, 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.

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

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.

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.

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

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

3 Estado del arte

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 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.

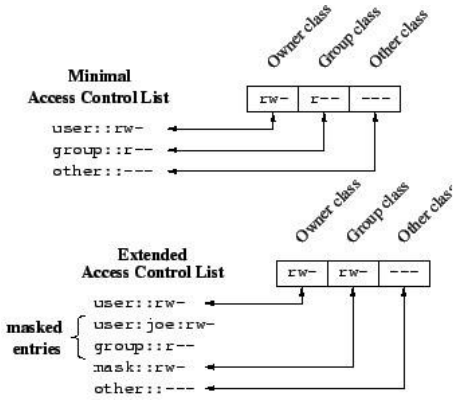

Figure 1: Access Control List

3.2 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.

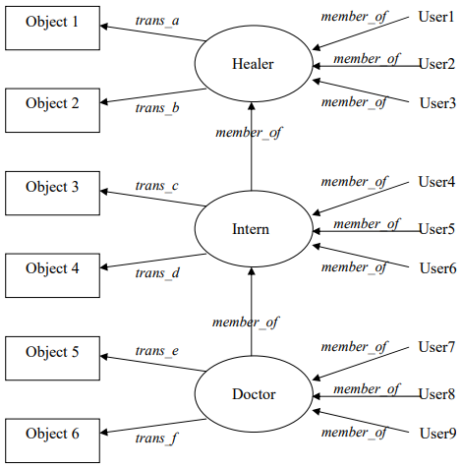

Figure 2: Role Based Access Control

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.

3.3 Pluggable Authentication Module