

# Doble factor de autenticación independiente de SO

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

May 2, 2018

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Motivación</b>                                | <b>1</b> |
| <b>2</b> | <b>Estado del arte</b>                           | <b>1</b> |
| 2.1      | Hardware . . . . .                               | 1        |
| 2.2      | software . . . . .                               | 1        |
| 2.2.1    | Usblock [Ste13] . . . . .                        | 1        |
| 2.3      | <i>Pluggable Authentication Module</i> . . . . . | 2        |
|          | <b>Bibliografía</b>                              | <b>4</b> |

### **Abstract**

El siguiente documento contiene el desarrollo de mi Trabajo de fin de grado el cual se centra en proporcionar una forma de bloquear o desbloquear un ordenador con arquitectura basada en Gnu is Not Unix/Linux sin necesidad de contraseña. Para conseguir esto, voy a usar un demonio que reside entre el sistema operativo y el Kernel. Al ser de tan bajo nivel, este programa, en principio debería ser independiente de sistema operativo. El único requisito es que el ordenador utilice Dbus [Ala13].

# 1 Motivación

Durante el transcurso del grado me he visto cada vez más interesado en la arquitectura GNU/Linux. A pesar de que no he impartido asignaturas orientadas a estos sistemas, mi propio interés y la base proporcionada por la carrera me han incitado a entrar en el mundo de GNU/Linux. Durante el transcurso de la carrera, he pasado de Ubuntu hasta Arch, profundizando poco a poco en el funcionamiento de estos sistemas GNU/Linux. Este trabajo representa la comprensión y desempeño que he adquirido a lo largo de la carrera.

Lo que quiero conseguir es implementar un Doble factor de autenticación o DFA en mi ordenador de forma que pueda bloquearlo si extraigo un USB específico. Una vez conseguido esto, haré la operación inversa. Conseguir desbloquear mi ordenador insertando un USB específico.

Para finalizar, proporcionar una interfaz para decidir que quiere habilitar o deshabilitar el usuario del programa. La herramienta en la que me voy a enfocar para hacer esto bien va a ser dbus, una aplicación que crea un protocolo de comunicación que pueden usar los programas del ordenador para comunicarse entre ellos.

# 2 Estado del arte

He encontrado varios métodos de desbloquear/bloquear el ordenador, los he catalogado en Hardware y Software.

## 2.1 Hardware

Estos dispositivos sirven para asegurar que el usuario es realmente quien tiene que ser. Existen varias marcas que ofrecen el mismo servicio, pero yo busco el original.

## 2.2 software

Para buscar proyectos parecidos a lo que quiero hacer, he buscado en GitHub. Muchos de los proyectos no cumplen con las especificaciones que quiero implementar en mi proyecto.

### 2.2.1 Usblock [Ste13]

Este proyecto consigue lo que yo quiero hacer, el problema es que no ha sido actualizado y por tanto ha quedado deprecado. El hecho de que use la capa *Hardware Abstraction Layer* hace que la mayoría de sistemas nuevos, por no decir todos van a ser incompatibles con este programa. Esta capa ha sido reemplazada por Udev [Gre07].

### ***2.3 Pluggable Authentication Module***

Este proyecto es un adaptador entre los nuevos protocolos de inicio de sesión y los programas que los utilizan. De esta forma, el desarrollador del programa no tiene que actualizar el software que ya ha escrito para dar soporte al nuevo método de autenticación, simplemente implementa *Pluggable Authentication Module*. La documentación incluye guías para administradores de sistemas y programadores.

## Glossary

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

**Doble factor de autenticación o DFA** Mayormente usado en cuentas on-line para verificar que realmente es la persona indicada quien accede a su cuenta. Consigue esto haciendo a la persona que se autentica en la cuenta pasar una prueba distinta a la contraseña, como verificación con SMS o por código autogenerado.

**Gnu is Not Unix** Proyecto con la finalidad de crear un sistema Unix-like completamente libre: GNU.

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

**Hardware Abstraction Layer** Capa que se encarga de gestionar los nuevos dispositivos encontrados en el sistema para que el usuario no tenga que gestionarlo él mismo. Si no encuentra el driver, pregunta al usuario del sistema.

**kernel** Intermediario entre el hardware y el software. Se encarga de gestionar la comunicación entre los programas y el hardware, de forma que si un programa quiere bajar el volumen de los altavoces, el programa se lo solicita al kernel y este al hardware mediante drivers..

**Linux** Kernel creado por Linus Torvalds.

**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.

**udev** Proporciona un directorio dinámico de dispositivos en el que elimina o añade nodos según los dispositivos que estén conectados. Generalmente se encuentran en /dev.

## Bibliografía

- [Gre07] Kay Sievers Greg Kroah-Hartman Dan Stekloff. *udev - Linux configurable dynamic device naming support*. July 2007. URL: <https://linux.die.net/man/8/udev>.
- [Ala13] Joe Rayhawk Alan Coopersmith. *Introduction to Dbus*. July 2013. URL: <https://www.freedesktop.org/wiki/IntroductionToDBus/>.
- [Ste13] Sven Steinbauer. *Lock and unlock your desktop using a USB key as a key*. Feb. 2013. URL: <https://github.com/Svenito/usbblock>.