Doble factor de autenticación independiente de SO

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

June 20, 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.

List of Figures

Abreviaciones y tecnicismos

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

bug Un bug, es un fallo en la línea de ejecución de un programa. ya sea lógico o sintáctico..

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.

Graphical User Interface Interfaz de usuario para comunicar el usuario con el ordenador mediante un conjunto de imágenes y objetos gráficos.

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.

National Institute of Standards and Technology Instituto que forma parte del departamento de comercio de Estados Unidos.

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

3.1 Evolución de las tecnologías

Los ordenadores forman una gran parte de nuestra sociedad desde que fueron inventados. El primer ordenador que se creó fue el Mark I, en 1944. Este ordenador estaba hecho mayoritariamente para realizar cálculos, pesaba cinco toneladas y se sobrecalentaba. Desde Mark I, como podemos observar en la imagen 1, los ordenadores han ido duplicando el número de transistores en circuitos integrados cada dos años [Moo98]. Hasta ahora, y seguramente unos años más en adelante, podamos ver que se sigue cumpliendo la ley de Moore.

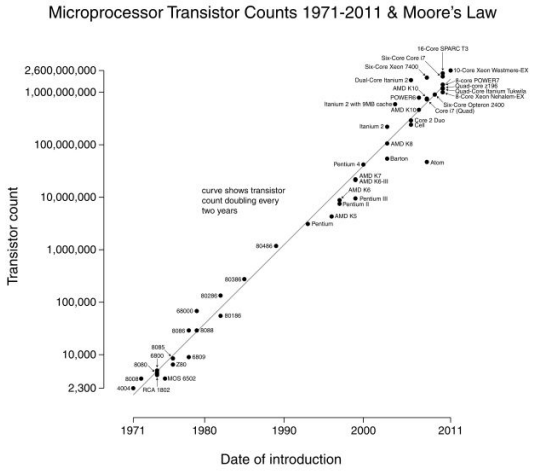

Figure 1: Ley de Moore

Durante el periodo de evolución de la informática, se pueden diferenciar cinco generaciones:

- **Primera generación:** Estos ordenadores no se parecían en nada a los ordenadores modernos, eran extremadamente grandes y poco sofisticados. Eran estructuras tan grandes como una habitación entera y usaban válvulas termodinámicas como switches y amplificadores. Las válvulas termodinámicas generaban mucho calor así que a pesar de contar con unidades de refrigeración enormes, se sobrecalentaban muy frecuentemente.
  Para interaccionar con estos ordenadores, los programadores usaban un lenguaje conocido como ”Lenguaje de máquina” que era directamente interpretable por el circuito.
  Esta generación tuvo lugar entre los años 1940 y 1956

- **Segunda generación:** Estos ordenadores se distinguen de la primera generación porque consiguen sustituir las válvulas termodinámicas por el transistor. El cambio

de válvulas a transistor implica más velocidad de cálculo y menos energía residual el forma de calor. Otra de las ventajas de los transistores es que, al ser más pequeños, reducen el tamaño del ordenador y lo hacen más económico.
En esta generación, también se consiguió almacenar información en discos, siendo esta la primera forma de almacenamiento de información persistente.
Esta generación tuvo lugar entre los años 1956 y 1963.

- **Tercera generación:** Se implementaron por primera vez los circuitos integrados, que contenían muchos transistores en chips semiconductores. Las ventajas de esta aportación fueron velocidad de cálculo, que aumentó mucho, el tamaño se redujo considerablemente y se continuaron abaratando los precios. En vez de "Lenguaje máquina", ahora los programadores usan monitores y teclados para comunicarse con el ordenador. Hasta esta generación, los ordenadores no estaban destinados para un uso personal. Solo las grandes empresas podían permitirse tener un ordenador.
  Esta generación tuvo lugar entre los años 1964 y 1971.

- **Cuarta generación:** Es la generación con más impacto en la sociedad. La tecnología avanzó hasta tal punto que los fabricantes podían poner miles de transistores en un único circuito integrado. EN esta época se puso a la venta el Intel 4004, el primer microprocesador en venderse de forma masiva. Este desarrollo inició la industria de los ordenadores personales.
  A mediados de los 70, salieron al mercado ordenadores como el Altair 8800 que venían a piezas y su usuario tenía que construir para usar. A finales de los 70, inicios de los 80, salieron al mercado ordenadores construidos de fábrica, como el Comodore pet, Apple II y el primer ordenador IBM. Al inicio de los 90, los ordenadores personales y la capacidad de crear una red de comunicación entre ellos dio paso a la creación de Internet. Se mejoró también la capacidad de almacenamiento de los discos además de la velocidad en general. Aparecieron lor primeros ordenadores portátiles, que presentaban una Graphical User Interface para facilitar la interacción entre el usuario y la máquina.
  Los usuarios de los ordenadores ya no tenían que ser gente técnica ni debía realizar un curso de formación para saber usarlos. Esto avivó la tasa de adopción de esta tecnología enormemente.
  Esta generación tuvo lugar entre los años 1971 y 2010.

- **Quinta generación:** Actualmente nos encontramos en esta generación. Algunos dicen que la adopción de los ordenadores cuánticos es el próximo gran paso, pero desde mi punto de vista, aún queda bastante para que eso ocurra.
  Creo que la quinta generación se caracterizará por el cambio de mentalidad de los usuarios de ordenadores. Hoy en día, tener la información guardada en un ordenador ya no es un privilegio, ni una ventaja competitiva. La verdadera ventaja es la disponibilidad de la información en cualquier ordenador al que te conectes, proporcionando al usuario una forma de acceso a información revolucionaria. Nunca antes ha sido posible acceder desde cualquier sitio a la información de un usuario.
  Esta generación empezó en 2010.

3.2 Consecuencias de la evolución

El rápido desarrollo de esta tecnología, como era de esperar cuando las cosas se hacen rápido y a contra reloj, introdujo un factor de error en la ecuación. Desde los inicios, ya sea por fallo humano o casualidad, se habla de un fenómeno llamado *Bug*. Este término forma parte de la jerga informática desde 1947, año en que, durante el ensamblaje del ordenador *Harvard Mark II*, tras el incorrecto funcionamiento del ordenador, los ingenieros revisaron las conexiones del ordenador y se encontraron con un insecto que adherido a dos cables, provocaba el fallo en el sistema. Desde entonces, se llama *Bug* al fallo en un programa, ya sea lógico o sintáctico.
Debido a la rápida adopción de los ordenadores por la población se continuó produciendo software con *Bugs*. Un estudio realizado por el *National Institute of Standards and Technology* concluyó en que los fallos en el software le cuestan a la economía estadounidense 59.5 billones de dólares anuales.[NIS02]
A medida que más usuarios usan programas, se van encontrando nuevos fallos, que dependiendo de la escala de gravedad podrían ser críticos, tanto para las empresas como para sus usuarios ya que si un usuario con malas intenciones encuentra un fallo de seguridad en la aplicación, dependiendo de su gravedad, podría, en teoría obtener información sensible de otros usuarios además de información interna de la compañía que ha hecho el software.
Para combatir este problema, algunas empresas han decidido recompensar a los usuarios que encuentran estos fallos. Al ofrecer una recompensa económica, la empresa incita al usuario a describir el fallo para que se pueda arreglar. Este tipo de programa se llama Bug Bounty Program. Existen varias páginas que se dedican a gestionar las ofertas de las empresas y los hallazgos de los usuarios para que ambos salgan ganando.
Hoy en día, tenemos todos nuestros datos *on-line*. Esto nos ofrece grandes ventajas como el acceso inmediato a nuestra información personal pero corremos un gran riesgo al confiar en las empresas que hacen que esto sea posible porque no existen programas sin *Bugs*.

3.3 Necesidad de seguridad

A medida que ha ido evolucionando la tecnología, medidas de seguridad previamente válidas, han ido quedando deprecadas debido a fallos que se han encontrado en los protocolos o nuevas versiones de estas mismas. El campo de la seguridad en la informática ha ido evolucionando como si se tratara del juego del ratón y el gato en el que los desarrolladores arreglan e inventan nuevas formas de proteger la información de los usuarios y los hackers vulneran esas implementaciones.
Ahora mismo, los usuarios son los que más tienen que perder. Tenemos todos nuestros datos almacenados en servidores de grandes empresas como Google o Facebook y en el caso de que se filtre nuestra información al mundo, tanto fotos personales como documentos sensibles se verían expuestos a todos los usuarios de Internet.
Para evitar esto, estas empresas implementan métodos de seguridad cada vez más avanzados como el doble factor de autenticación para iniciar sesión en la cuenta.
Las medidas de seguridad que proporcionan las empresas como Google o Facebook son

bastante buenas pero no son suficientes. Tampoco podemos pedir a estas empresas que implementen todo tipo de sistemas de autenticación de usuarios, ya que al final, tienen que hacer el proceso fácil y sencillo para que le gente quiera y sepa usarlo.
Podemos aumentar la seguridad de nuestro sistema configurando nosotros mismos las distintas formas de autenticarnos en nuestros sistemas. Una de las formas en las que podemos aumentar la seguridad de nuestro sistema es mediante el Doble Factor de Autenticación. Algunos ejemplos de esta implementación son:

- Contraseña + Llave USB
- Contraseña + App generadora de códigos (Google Authenticator)
- App generadora de códigos + Sensor de huella dactilar
- Contraseña + Sensor de retina

A cuanto más valor, ya sea económico o emocional, más medidas de seguridad serán implementadas en proteger dicha información.
En definitiva, es muy complicado tener un entorno seguro en el que poder confiar porque muchas de los vectores de ataque no dependen del usuario final, sino de terceros (Tantos como distintos servicios use el usuario). Es importante tener un buen nivel de seguridad en todos los dispositivos, pero este no debe interponerse entre las tareas que un usuario tiene que hacer en su ordenador ya que de lo contrario, sería un inconveniente, no una ventaja. Cada usuario debe establecer el grado de seguridad de sus cuentas, tanto en Internet como de sistema.
Creo que existe una relación fuerte entre las medidas de seguridad que implementan los usuarios y el valor de la información que esa información tiene para ellos. Para la mayoría de usuarios, una contraseña, aunque menos segura que un Doble Factor de Autenticación será más conveniente porque la naturaleza de la información que tiene que proteger no es tan importante.

4 Introducción

Supe desde el primer momento en que nos dijeron que fuésemos pensando sobre que queríamos hacer el trabajo de fin de grado que quería hacerlo sobre GNU/Linux. Como todo alumno de primero de carrera, inicié el curso con ganas y Windows.
En primero ya me di cuenta de que no era el mejor sistema operativo para desarrollar. Cada vez que usaba una máquina virtual para hacer cualquier práctica, me planteaba cambiarme a Ubuntu y dejar atrás Windows.
Al empezar segundo de carrera, decidí hacer una partición para Ubuntu. De esta forma, no tenía que iniciar una máquina virtual en Windows para trabajar. Durante dos años estuve usando este sistema pero no estaba contento porque sabía que no era mi sistema operativo.
Durante el verano del tercer año en la carrera decidí que quería cambiar a Arch linux

pero vi varios comentarios en foros que sería mejor pasar antes por Manjaro para acostumbrarme al ecosistema de Arch linux y coger soltura y confianza usando comandos a diario.
Entrando en el segundo cuatrimestre de cuarto (Quinto año en la carrera) me sentía lo suficientemente cómodo como para instalar Arch linux en mi sistema.
Todo el proceso de aprendizaje que he pasado durante estos años me ha servido para aprender y valorar el sistema operativo.
Trás toda esta experiencia, he querido ampliar mis conocimientos en GNU/Linux. Durante la reunión con mi tutor, vimos algunas opciones de investigación y finalmente, al estar interesados en la seguridad de las aplicaciones y sistemas, decidimos que era buena idea tirar por esa rama.

5 Objetivos

6 Métodos de trabajo

7 Investigación y resultados

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.

7.1 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.

7.1.1 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.

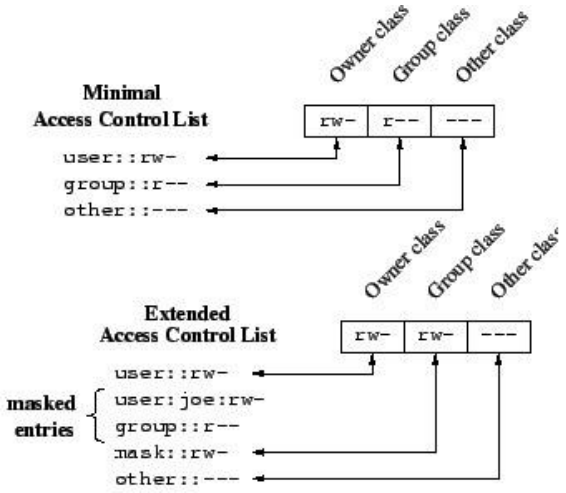

Figure 2: Access Control List

7.1.2 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 3 se puede ver claramente como dependiendo del rol vas a poder acceder a ciertos objetos.

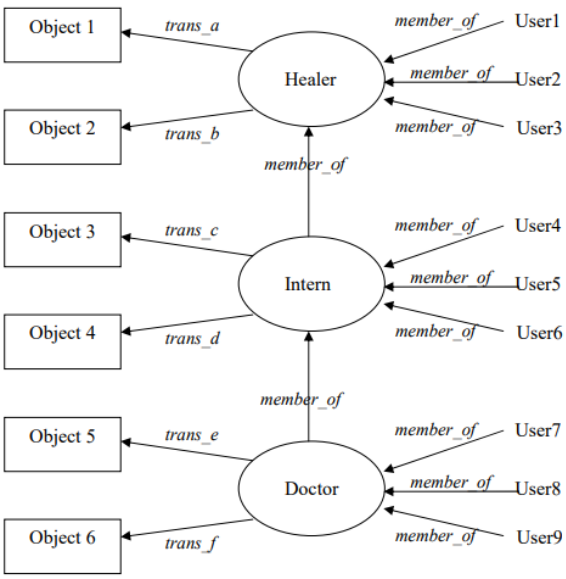

Figure 3: Role Based Access Control

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.

7.1.3 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.

7.1.4 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.

7.1.5 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.

7.1.6 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.

7.1.7 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.

7.1.8 Pluggable Authentication Module

8 conclusiones

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/.

[Moo98] Gordon Moore. *Cramming More Components onto Integrated Circuits*. Jan. 1998. URL: http://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf.

[NIS02] NIST. *Software Errors Cost U.S. Economy* 59.5*BillionAnnually*. Oct. 2002. URL: https://web.archive.org/web/20090610052743/http://www.nist.gov/public_affairs/releases/n02-10.htm.