Capítulo 9

Conclusiones

Estado del arte

Posibles ampliaciones

Capítulo 9 Conclusiones

Por último, y como colofón al proyecto, vamos a comentar los resultados obtenidos y las posibles mejoras.

Expondremos el estado del arte de la integración de la autenticación, y describiremos un caso real y cercano en el que se utiliza de forma indispensable. Se trata de los centros educativos TIC de Andalucía.

9.1 Estado del arte

El problema de la autenticación en sistemas de muchas computadoras ha sido resuelto de múltiples maneras a lo largo del tiempo. Una de las formas que más se ha usado ha sido la utilización de los mapas NIS.

NIS permite administrar en red los archivos de sistema más importantes, como por ejemplo el archivo de contraseñas /etc/passwd. Se instala un cliente en cada máquina que obtiene estos ficheros de un servidor NIS y poco más.

Existen algunos problemas al trabajar con estos ficheros. Si intentamos cambiar una contraseña de la forma habitual (mediante passwd), la próxima vez que entremos en el sistema nos encontraremos con la que teníamos en un principio. No se ha cambiado puesto que passwd hacía referencia al fichero de claves local y no al que existe en el servidor, y que se trasmite a todos los clientes. Para lograr el cambio de contraseña necesitamos una herramienta adicional, yppasswd.

Además de yppasswd existen más herramientas (pertenecientes al conjunto yp-tools) que nos permiten trabajar con los mapas NIS. Y aunque esto parezca un poco engorroso, no es el principal problema. Ya hemos hablado bastante sobre la seguridad, y todo lo que se ha dicho es válido para este caso.

NIS no tiene soporte para seguridad. Como solución a esto apareció NIS+, pero que ya es mucho más difícil de administrar, y que tardó en ser estable en Linux.

Ante esta situación surge la idea de mantener la información necesaria para autenticación en una base de datos. Esta información suele cambiar

poco por lo que se eligió usar un directorio, que no es más que una base de datos optimizada para lectura.

Por otro lado, el sistema debe ser centralizado. Toda la información se encuentra en el directorio y todos los usuarios se autentican contra este directorio. Todo esto facilita en gran medida la administración.

Ya no existe el problema de distribuir los ficheros de claves completos, aunque sí se van a transmitir las claves. Este problema se solventa fácilmente mediante el uso de una capa segura TLS.

Al igual que ocurría con NIS, no sólo se almacenan contraseñas sino que el directorio también admite otro tipo de información, tal como nombres de equipos en la red, información personal de usuario…

Conforme crece el número de usuarios del sistema, el servidor tiene que dar repuesta a un mayor número de peticiones, y es probable que tienda a consumir muchísimos recursos. Existen diversas formas de aumentar el rendimiento del sistema. De hecho vamos a comentado varias en el apartado de ampliaciones.

Mediante una serie de pequeñas modificaciones, podemos hacer que el sistema base dé servicio a un número bastante alto de usuarios (entorno a 10.000).

Un ejemplo claro y cercano de organización en la que se utiliza un sistema con autenticación integrada (entre otros servicios) es la red de centros educativos TIC de Andalucía.

En cada centro TIC cuentan con una serie de equipos conectados a un servidor, que hace las veces de cortafuegos, por el que se conectan al exterior.

En cuanto a la autenticación los alumnos pueden entrar en cualquier equipo con su nombre de usuario y clave. Además al acceder al equipo importan automáticamente su directorio Home desde el servidor (usando para ello NFS). De esta forma también llevan consigo sus perfiles, documentos y demás.

9.2 Posibles ampliaciones del proyecto

A continuación sugerimos ciertas mejoras con las que el presente proyecto se vería muy complementado. Algunas de ellas pueden considerarse anecdóticas, otras por el contrario son muy necesarias para el correcto funcionamiento de un sistema con una cantidad enorme de usuarios.

Vamos a exponerlas de forma que queden ordenadas de mayor a menor importancia.

9.2.1 Optimizar el directorio

Podemos acelerar el acceso al directorio de distintas formas:

- Mediante una correcta configuración de los atributos de índice (index). Sólo debemos usarlos para aquellos atributos que vayamos a usar para realizar búsquedas.

- Deshabilitando la creación de archivos de log (loglevel 0). Este es un arma de doble filo pues aunque el ahorro de ciclos de procesador y de memoria puede llegar a ser considerable, perdemos el control de lo que realiza el servidor.

- Optimizar la memoria caché. Debemos ajustar los parámetros cachesize y dbcachesize a nuestro sistema.

- Gestión correcta de los usuarios. Se debe diseñar un árbol jerárquico consecuente con la jerarquía de la organización a la que pertenece.

- Configuración del demonio nscd en las máquina cliente. Conforme crece el número de usuarios, la carga de la red se vuelve considerable. Podemos reducir ancho de banda y carga en el servidor mediante nscd. Este

demonio cachea la información de usuario de tal forma que no se debe recurrir al directorio cada vez que demande esta información.

9.2.2 Recuperación ante desastres

Aunque OpenLDAP no se integra demasiado bien con las aplicaciones comerciales de backup, disponemos de otras herramientas para tal labor. Se trata del sistema de réplica. Nos permite tener varios servidores (a los que se conectan las aplicaciones) de tal forma que si falla uno, podemos tener otro operativo con todos los datos.

Para replicar un directorio con OpenLDAP debemos elegir un servidor como maestro, pasando el resto a ser esclavos de éste. En el servidor maestro se configura el demonio slurpd, que se encarga de realizar las replicaciones.

Tras una sencilla configuración y a partir de ese momento, cada cambio que se realice en el servidor maestro será replicado a todos los esclavos.

Dada la imposibilidad de realizar un backup "en caliente" de un servidor OpenLDAP, el método más común de realizarlo es usar un script que en un determinado momento:

- Parar el servidor (slapd y slurpd)
- Hacer una copia de la base de datos y de la configuración.
- Comprimir la copia y dejarla accesible a nuestra aplicación de backup.
- Arrancar el servidor y comprobar que todo funciona correctamente.

A la hora de recuperar un servidor bastaría con:

- Instalar la misma versión de OpenLDAP que tenía el servidor que ha fallado.

- Copiar la base de datos y el archivo de configuración.

- Arrancar el servidor.

9.2.3 Uso de NFS para importar el directorio home

Mediante el uso de NFS (Network File System) podemos conseguir una total independencia del equipo de la red desde donde trabajemos. No sólo podremos acceder a cualquier equipo, entrar en la web o consultar el correo, sino que además podemos disponer de nuestro perfil de usuario, archivos y demás.

Al acceder al sistema el directorio home se importará desde un servidor habilitado para tal efecto mediante NFS. A partir de ese momento trabajará con ese directorio como si fuese local. En él se almacenarán los archivos ocultos que mantienen la configuración de usuario de cada programa, las cookies incluso los archivos que el usuario desee conservar.

Cada vez que se producen cambios en el directorio home se almacenan en el servidor. Al terminar la sesión se desmonta el directorio.

9.2.4 Uso de Debian como sistema operativo

Uno de los requisitos que encontramos a la hora de realizar el proyecto era usar la distribución Red Hat Linux 9.0. Los servidores en los que se instalaría el sistema desarrollado usaban este sistema operativo.

Esta elección trajo diversas consecuencias. La principal fue la obligación de compilar la mayor parte de los paquetes que se utilizan. Esto es así

debido a que la empresa Red Hat Inc. dejó de dar soporte al sistema operativo. Dejó de sacar actualizaciones de sus paquetes.

En general siempre tendemos a instalar la versión más nueva de cada paquete. Algunas veces, tal como ha ocurrido durante el desarrollo del proyecto, esta decisión no se toma de manera arbitraria. Existen diversos paquetes cuya instalación exige una determinada versión de otro paquete. Esto se conoce como dependencias.

En esta ocasión algunas de las dependencias de OpenLDAP no existían de forma binaria (rpms) por lo que era inevitable compilar desde fuente.

Este motivo no es el único que nos hace pensar en migrar a Debian. El hecho de que Red Hat no ofrezca más actualizaciones es bastante grave. Existen muchos paquetes, sobre todo los relacionados con la seguridad, que deben estar al día. Conforme se dan a conocer agujeros de seguridad el administrador debe taparlos, si quiere seguir disfrutando de una red segura.

Las actualizaciones de Debian dependen de una comunidad de voluntarios y no de una empresa, por lo que no vamos a perder el soporte de la noche a la mañana, tal como nos ha sucedido con Red Hat. Además en Debian disponemos de numerosas herramientas que facilitan en gran parte la tarea del administrador. Cuenta, por ejemplo, con la herramienta apt, que nos permite instalar o actualizar paquetes de forma sencilla y resolviendo las dependencias.

Para la implementación en Debian del sistema desarrollado, habría que estudiar en primer lugar que versión se va a utilizar. En el proyecto Debian se mantienen activas tres versiones: estable, en pruebas e inestable. Se recomienda siempre usar la estable, que en estos momentos se denomina Sarge, aunque a veces es necesario recurrir a la versión en pruebas. Los

paquetes en esta versión están actualizados y corregidos de la mayor parte de errores o bugs.

Una de las primeras tareas que debemos realizar es estudiar la posibilidad de instalar los paquetes desde binarios. Se tendría que analizar qué versiones son las necesarias, y cuales son las que disponemos, tanto de paquetes como de librerías.

En principio no debería haber demasiados problemas, quizás qmail sea el único paquete que habría que compilar desde fuente (debido a su licencia). Si es posible se seguiría este camino, puesto que ahorra trabajo y tiempo.

Los pasos a seguir serían los mismos, instalación, configuración y pruebas.

La instalación sería ordenada. En primer lugar habría que desinstalar los paquetes relacionados que ya se encuentren en el sistema, eliminando los ficheros de configuración. Esto es, deben ser desinstalados usando la opción --purge de la herramienta apt-get.

En cuanto a la configuración, los ficheros en ambas plataformas son idénticos, para las mismas versiones, si bien puede diferir en el contenido, de forma puntual. Además la localización de éstos suele ser distinta. Al instalar desde fuente hemos elegido, de una forma u otra, el path de cada uno de ellos.

En el caso de que hubiese que instalar distintos paquetes habría que estudiar el apartado de configuración en ellos también.

La fase de pruebas del sistema sería idéntico en Debian. Debemos asegurarnos de que el sistema funciona como se espera, comprobando cada una de las aplicaciones.

9.2.5 Libretas de direcciones

Hasta ahora sólo hemos utilizado el directorio para almacenar la información relativa a la autenticación del usuario. El nombre, la contraseña cifrada, el directorio home y pocas cosas más.

Podemos añadir también cierta información personal: nombre completo, dirección, teléfono, incluso fotos.

De esta forma toda la información estaría centralizada y sería fácil de administrar. Cualquier aplicación de tipo agenda o libreta de direcciones que tuviese soporte LDAP podría acceder a estos datos.

Recordemos el problema de no disponer de unos datos de contacto actualizados. Mediante LDAP quedaría resuelto.

9.2.6 Instalación y configuración de phpLDAPadmin

Existen varias formas de gestionar la información que contiene el directorio. Una de ellas es la que hemos venido utilizando hasta ahora. Usábamos los comandos que nos proporcionaba el paquete LDAP y a veces programamos algunos scripts.

Una de las formas más fáciles de gestionar el directorio es hacer uso de una herramienta gráfica. En esta ocasión proponemos phpLDAPadmin con la que podemos inspeccionar la estructura jerárquica del directorio y actuar consecuentemente de manera rápida.

Esta aplicación cuenta con multitud de características, lo que permite hacer prácticamente cualquier tarea administrativa y de mantenimiento.

Incluso tiene ventajas para el tratamiento masivo de información, campo en el que podríamos pensar que dominan los scripts.

9.2.7 Mejoras en los servicios integrados

Es posible pensar que esta parte no está totalmente relacionada con el objetivo de nuestro proyecto. Hasta ahora hemos estudiado por encima cada servicio, para después acometer su integración. Esto era necesario puesto que el servicio en cuestión debía funcionar correctamente, no sólo para la autenticación.

En este apartado proponemos algunas mejoras de estos servicios que pueden llegar a ser importantes, tanto a nivel de la autenticación, como a nivel general de funcionamiento.

Squid

- Hacer uso de la potencia que las ACL nos proporcionan
- Instalar Dansguardian como filtro de contenidos y adecuar las reglas a la organización que se administra.

Postgresql

- Instalar y configurar una herramienta gráfica de gestión, como por ejemplo phpPgadmin

qmail

- Dar la posibilidad de acceso pop al usuario, e integrar la autenticación en este caso
- Dar la posibilidad de cifrar los correos mediante SSL
- Instalar y configurar una herramienta gráfica de gestión, como por ejemplo qmailadmin
- Configurar las pantallas de SqWebMail para adecuarlas a la organización