MKPolohomedir. Módulo de PAM para la gestión de usuarios

Diego Martín Arroyo

1 de mayo de 2015

Índice

Introducción

La gestión de los usuarios debe ser realizada de forma transparente. Permitiendo a los usuarios acceder al mismo en cualquier nodo y sin ningún tipo de configuración necesaria por su parte, acceder a todos los recursos presentes en el resto de nodos.
El sistema delega parte de la gestión de los usuarios a un directorio **LDAP**, gestionando el acceso al sistema mediante el módulo **PAM**. Sin embargo, debido a las necesidades particulares del sistema, la configuración básica de **PAM** y las herramientas incluidas no son suficientes para conseguir el funcionamiento deseado, y por ello es necesario desarrollar un conjunto de herramientas propias.

Necesidades

El sistema se compone de una serie de nodos independientes a los cuales pueden acceder los usuarios indistintamente. Se desea que los usuarios puedan utilizar los recursos presentes en cualquier nodo sin tener que acceder directamente a ellos (mediante una sesión remota, por ejemplo), y puedan empezar a trabajar sin tener que realizar ningún tipo de configuración.
Esta necesidad está parcialmente cubierta por los módulo de PAM **pam_unix.so** (gestión de usuarios locales)[1], **pam_ldap.so** (gestión de usuarios del servidor **LDAP**)[2] y en particular **pam_mkhomedir.so**[3], que permite la creación de un directorio de usuario si el mismo ha iniciado sesión por primera vez en esta máquina.
Sin embargo, una de las características del sistema es la uniformidad de sus nodos en cuando a usuarios se refiere. Por ello, es deseable que los usuarios cuenten con un conjunto básico de datos en cualquier nodo del sistema a la hora de acceder por primera vez. El módulo **pam_mkhomedir** no lleva a cabo esta tarea, pues únicamente crea el directorio (realizando una copia de `/etc/skel` de inicio del usuario en la máquina donde se ha realizado el acceso).
Es por tanto necesario crear un mecanismo para llevar a cabo dicha tarea. Aprovechando la funcionalidad de PAM, se ha procedido a crear un módulo complementario a **pam_mkhomedir** denominado **pam_mkpolohomedir**, que aprovecha la herramienta **MarcoPolo** para realizar su cometido.

Características del módulo

La funcionalidad de un módulo de **PAM** se recoge en un objecto compartido que se vincula en tiempo de ejecución con el resto de componentes de **PAM** según se disponga en los ficheros de configuración del módulo (generalmente situados en `/etc/pam.d/`). [4] y que se debe almacenar en `/lib/security`.
En el caso particular a resolver, el módulo será invocado tras la ejecución del módulo **pam_mkhomedir**, y se encargará de, aprovechando la funcionalidad de MarcoPolo, detectar todos los nodos disponibles en la red y proceder a la creación en los mismos del directorio de inicio, así como la realización de una serie de tareas adicionales.

Toda la funcionalidad se recoge en el fichero `pam_mkpolohomedir.c`. Dicho módulo recoge la información de interés (nombre, uid y gid del usuario) a través de llamadas al núcleo de **PAM**[5].

Listing 1: Obtención de la información del usuario

```
int pam_sm_open_session(pam_handle_t * pamh, int flags, int argc
            ,const char **argv)
{
   int retval, ctrl;
   const char *user;
   const struct passwd *pwd;
   struct stat St;

   /* Parse the flag values */
   ctrl = _pam_parse(flags, argc, argv);

   /* Determine the user name so we can get the home directory */
   retval = pam_get_item(pamh, PAM_USER, (const void **) &user);
   if (retval != PAM_SUCCESS || user == NULL || *user == '\0')
   {
      _log_err(LOG_NOTICE, "user unknown");
      return PAM_USER_UNKNOWN;
   }
```

Una vez obtenida dicha información, las funciones `create_polo_homedir` y `createdirs` se encargan de la búsqueda de los nodos del sistema (aprovechando el *binding* de **Marco**) y de contactar con ellos.

Creación del directorio

Cada nodo cuenta con una instancia del esclavo de **polousers**, el servicio que se encarga de recibir las peticiones de creación de usuarios. Dicho servicio se implementa sobre el *framework* **Twisted**, y verifica la autoría de cada petición mediante sockets sobre TLS (*Transport Layer Security*).
Una vez creado el directorio, el servicio así lo indica al solicitante, que almacena el registro de dicha operación.

Referencias

[1] *pam_unix(8) - Linux man pages.*

[2] *pam_ldap(5) - Linux man pages.*

[3] J. Gunthorpe, *pam_mkhomedir(8) - Linux man pages.*

[4] A. G. Morgan and T. Kukuk, *The Linux-PAM Module Writers' Guide.* linux-pam.org, 1.1.2 ed., Aug. 2010.

[5] A. G. Morgan and T. Kukuk, *The Linux-PAM Module Writers' Guide*, ch. 2. What can be expected by the module. In [4], 1.1.2 ed., Aug. 2010.