Mecanismos de autenticación por contraseña

Alberto Molina Coballes

IES Gonzalo Nazareno
CONSEJERÍA DE EDUCACIÓN

14 de noviembre de 2011

Introducción

- En el proceso de autenticación de un usuario en el sistema, este último utilizará algún mecanismo para asegurarse de que el usuario es quien dice ser.
- Se pueden utilizar métodos muy diversos: Aspectos biométricos (huella dactilar, iris, fondo de ojo, ...), claves criptográficas, tarjetas inteligentes, contraseñas, etc.
- El mecanismo más habitual es el uso de contraseñas, aunque la forma de almacenar estas contraseñas es muy diverso.
- En esta presentación veremos los principales mecanismos de autenticación por contraseña.

Índice

Contraseñas almacenadas en fichero

Contraseñas almacenadas en claro / hash

- Una opción sería utilizar un formato del tipo:

  `usuario:contraseña`
- Cualquiera que pueda leer el fichero podrá ver la contraseña.
- Ese tipo de fichero se utiliza en ocasiones, pero es mucho más frecuente que no se almacene la contraseña en claro, sino su hash.
- Un ejemplo clásico sería una línea del fichero `/etc/passwd` de cualquier sistema UNIX:

  `oneadmin:mbjXwldxJRm6I:120:129::/var/lib/one:/bin/bash`

Hash en fichero (UNIX) - I

- Hoy en día no se almacenan las contraseñas en el fichero `/etc/passwd` sino en `/etc/shadow` que sólo puede leer `root`. Esto se denomina *shadow passwords*.
- En el segundo campo del fichero `passwd` se pone una `x` para indicar que la contraseña está en el fichero `shadow`.
- Los algoritmos de cifrado para el hash de las contraseñas han ido cambiando con el tiempo.

  DES Muy importante en su momento, pero hoy en día descartado por su debilidad. Ejemplo:
  `mbjXwldxJRm6I`

  NTLMv1 Utilizado por compatibilidad con Windows, pero no recomendado por su debilidad. Ejemplo (El prefijo $3 indica que se trata de NTLMv1):
  `$395jllEOiWm.`

Hash en fichero (UNIX) - II

- Se introduce la sal para aumentar la dificultad de romper la contraseñas.
- Los hashes de las contraseñas se codifican en base64 para que sean portables.
- Se utilizan diversos algoritmos con sal: MD5, Blowfish, SHA256 o SHA512, distinguiendo uno de otro por el prefijo. En el siguiente ejemplo $6 indica que se ha utilizado SHA512 y la sal utilizada se indica entre $ (yPpZhimP):

  ```
  $6$yPpZhimP$Pj/1looF9eGD//T8uwP22/EXWzhK03Nzcug7tFM\
  7ZM1.YdG25fMmQjO2fBuFD8Knvt7CkszonGDBI5xD/iDa30
  ```

- La mayor parte de las distribuciones GNU/Linux utilizan hoy en día el algoritmo de hash SHA512, durante bastantes años el más habitual fue MD5.

Hash en fichero (servicios)

- No sólo es necesario autenticarse en el sistema operativo, numerosos servicios solicitan autenticación a los usuarios.
- Los usuarios de los servicios pueden ser usuarios del sistema o sólo usuarios del servicio (usuarios virtuales).
- Los usuarios del servicio pueden tener su propio fichero de usuarios y contraseñas.
- Es frecuente que estos servicios utilicen un fichero con el mismo formato que el fichero `passwd`. Ejemplos: Apache, Squid o Proftpd
- Incluso tienen aplicaciones para generar los hashes (`htpasswd`).

Hash en el Registro de Windows

- Windows almacena las contraseñas de los usuarios en el registro Security Account Manager (SAM)
- Utiliza diferentes algoritmos de cifrado

  LM Lan Manager. Formato obsoleto que se mantiene en los sistemas Windows por compatibilidad con versiones anteriores, aunque está deshabilitado por defecto.

  NTLM Lan Manager de Windows NT. La versión 2 es la que se utiliza actualmente en los sistemas Windows.
- Los algoritmos utilizados tanto por LM como por NTLM son débiles (NTLM usa una variante de HMAC-MD5).
- El SAM puede protegerse opcionalmente con la aplicación SYSKEY, que cifra toda la base de datos SAM con una contraseña.

Contraseñas almacenadas en Bases de Datos Relacionales

Autenticación LDAP

Autenticación LDAP

- Se puede utilizar en sistemas centralizados.
- No confundir con autenticación Kerberos + LDAP
- Todos los equipos de una red (dominio en terminología Windows) pueden autenticar a los usuarios del directorio.
- Para almacenar la contraseña en el directorio activo de Windows se utiliza el atributo `unicodePwd` del objectClass `user`. La contraseña se almacena en claro, entrecomillada y en base64.
- Para almacenar la contraseña en un directorio sobre un sistema UNIX (openLDAP por ejemplo), se utiliza el atributo `userPassword` del objectClass `posixAccount`. Se puede almacenar la contraseña en claro o preferiblemente el hash de la misma.
- Incluir la información de autenticación de un usuario en LDAP permite que sea un sistema utilizado no sólo por los sistemas, sino también por un sinfín de aplicaciones (web, proxies, correo, etc.)

Ejemplo de usuario en LDAP (UNIX)

- Una entrada muy simple (con los atributos mínimos y la contraseña en hash MD5), sería:

```
dn: uid=usuario,ou=People,dc=example,dc=com
uid: usuario
cn: usuario
objectClass: account
objectClass: posixAccount
objectClass: top
userPassword: {MD5}qPXxZ/RPSWTmyZje6CcRDA==
loginShell: /bin/bash
uidNumber: 2000
gidNumber: 2000
homeDirectory: /home/usuario
```

Ejemplo de usuario en LDAP (Windows)

```
dn: CN=Piet Prutser,CN=Users,DC=forest,DC=example,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=example,DC=com
distinguishedName: CN=Piet Prutser,CN=Users,DC=forest,DC=example,DC=com
cn: Piet Prutser
sn: Prutser
givenName: Piet
displayName: Piet Prutser
name: Piet Prutser
instanceType: 4
userAccountControl: 514
accountExpires: 0
samAccountName: pprutser
userPrincipalName: P.Prutser@example.com
altSecurityIdentities: Kerberos:pprutser@EXAMLE.COM
unicodePwd:: IgBuAGUAdwBQAGEAcwBzAHcAbwByAGQAIgA=
```