

# Las contraseñas en Windows

Vte. Javier García Mayén

neofito(at)gmail(dot)com

<http://www.wadalbertia.org>

## 1. Principios de seguridad

En los sistemas `Windows` los controles de acceso a los recursos, entendiéndose por tales los archivos, carpetas y demás objetos, se aplican a los denominados principios de seguridad, incluyendo entre estos los siguientes:

- Usuarios
- Grupos
- Equipos

En principio solo nos interesan los primeros (las cuentas de usuario) por lo que nos centraremos en ellos.

## 2. Security Account Manager (SAM)

Un usuario habitualmente se identifica en `Windows` proporcionando el nombre de una cuenta y la contraseña asociada a la misma. Toda esta información, y alguna más, se almacena en el `Security Account Manager (SAM)`.

Las contraseñas, por supuesto, no se almacenan en texto claro, sino que en su lugar lo que allí encontraremos será el hash de las mismas. Por otra parte, y desde la aparición del `Service Pack 3` para `Windows NT` existe la posibilidad de añadir una nueva capa de cifrado a los hashes del `SAM`, utilizando para ello una clave aleatoria de 128 bits. Esta clave adicional se conoce como `SYSKEY`.

El `SAM` conforma una de las cinco ramas del registro de `Windows` (`HKEY_USERS`) y su ubicación varía en función de si el equipo pertenece a un dominio o por contra es un equipo independiente:

- En un dominio se ubica en los controladores de dominio en el archivo:

```
%systemroot%\ntds\ntds.dit
```

- En un equipo independiente se ubica en el archivo

`%systemroot%\system32\config\sam.`

En ambos casos el acceso a dicho fichero esta vetado durante la ejecución del sistema, por lo que no podremos ni copiarlo, ni borrarlo ni moverlo.

Quedan fuera del alcance de este documento los diferentes métodos que podemos utilizar para obtener/editar el SAM de forma offline, que haberlos haylos.

### 3. Almacenamiento de los nombres para las cuentas de usuario

Además de lo indicado anteriormente en la SAM la información referente a las cuentas de usuario se almacena utilizando un numero de 48 bits único denominado Identificador de Seguridad (SID).

Un ejemplo para el mismo podría ser el siguiente:

```
S-1-5-21-117609710-1343024091-842925246-500
```

En dicho número podemos diferenciar dos partes:

- La primera de ellas es única para cada instalación o dominio y se correspondería con S-1-5-21-117609710-1343024091-842925246 en el ejemplo anterior.
- La segunda parte es compartida por todas las instalaciones de Windows y se conoce como RID. Sería el 500 del ejemplo anterior.

Sirvan como ejemplo los siguientes números RID "bien conocidos" (para conocer el resto visitar el enlace mencionado en las Referencias):

- 500 → Administrador
- 501 → Invitado
- 512 → Administradores del Dominio
- 513 → Usuarios del Dominio

Por otra parte, a la primera cuenta de usuario creada en un sistema local o dominio se le asigna un RID igual a 1000, aumentando este en 1 para cada una de las subsiguientes.

### 4. Almacenamiento de los hashes para las contraseñas de los usuarios

Por defecto los sistemas Windows permiten contraseñas de una longitud máxima igual a 14 caracteres. En el SAM se almacenan dos versiones para cada una de las contraseñas:

1. Hash LanMan o LM (cifrado mediante el algoritmo DES).
2. Hash NT (cifrado mediante el algoritmo MD4).

Estos algoritmos de cifrado se dice que son unidireccionales ya que una vez cifrada la contraseña no puede realizarse el proceso inverso, es decir, a partir de un hash no puede obtenerse la contraseña original.

Entonces, ¿cómo funcionan los programas de cracking de contraseñas? Básicamente funcionan cifrando textos y comparando el hash obtenido con el de una cuenta determinada. Si ambos coinciden habremos dado con la contraseña original.

El método de generación de los textos puede ser mediante fuerza bruta (dado un conjunto de caracteres el programa genera de forma aleatoria todas las combinaciones posibles de los mismos) o un ataque por diccionario (listas de palabras proporcionadas a priori).

## 5. Particularidades de la generación del hash LanMan

Para generar el hash LM de una contraseña determinada Windows utiliza un método ampliamente conocido y que precisamente constituye la debilidad de este tipo de hashes.

Independientemente de la longitud de la contraseña original esta se divide en dos mitades de 7 caracteres cada una, se convierten a mayúsculas todos los caracteres y se generan los hashes de manera independiente. Esto implica que la generación de la segunda parte del hash no utiliza ninguna información relacionada con la primera mitad por lo que, a todos los efectos, estaríamos tratando con dos contraseñas de 7 caracteres.

Vamos a hacer algunas pruebas para comprobar esto. Para ello crearemos 3 usuarios diferentes y volcaremos los hashes de las contraseñas de cada uno de ellos estudiando a continuación los resultados obtenidos. Para realizar el volcado de las contraseñas deberemos contar con privilegios de administrador en la máquina utilizada para las pruebas. Obtendremos igualmente el SID de cada una de las cuentas.

Abramos una consola y creemos los usuarios con la siguiente información:

- Cuenta número 1:

Usuario: *wadalberto1*  
Contraseña: *wadalbertia*

```
C:\>net user wadalberto1 wadalbertia /add  
Se ha completado el comando correctamente.
```

- Cuenta número 2:

Usuario: *wadalberto2*  
Contraseña: *wadal1*

```
C:\>net user wadalberto2 wadal1 /add  
Se ha completado el comando correctamente.
```

- Cuenta número 3:

Usuario: *wadalberto3*  
Contraseña: *wadal2*

```
C:\>net user wadalberto3 wadal2 /add  
Se ha completado el comando correctamente.
```

Volcaremos a continuación la información relacionada con el SID de cada una de las cuentas anteriores. Para ello utilizaremos el programa `user2sid`:

```
C:\>user2sid.exe wadalberto1  
  
S-1-5-21-117609710-1343024091-842925246-1005  
  
Number of subauthorities is 5  
Domain is ANUBIS  
Length of SID in memory is 28 bytes  
Type of SID is SidTypeUser
```

```
C:\>user2sid.exe wadalberto2

S-1-5-21-117609710-1343024091-842925246-1006

Number of subauthorities is 5
Domain is ANUBIS
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser
```

```
C:\>user2sid.exe wadalberto3

S-1-5-21-117609710-1343024091-842925246-1007

Number of subauthorities is 5
Domain is ANUBIS
Length of SID in memory is 28 bytes
Type of SID is SidTypeUser
```

En los tres casos podemos comprobar como la información de las tres cuentas tiene los rasgos ya mencionados:

- La primera parte que compone el SID coincide en todos los casos. En mi maquina se correspondería con S-1-5-21-117609710-1343024091-842925246
- El RID para los tres es mayor que 1000 y los números son sucesivos. En mi caso: 1005, 1006 y 1007 (correspondientes, respectivamente, a la quinta, sexta y séptima cuenta de usuario creadas en el sistema).

Ahora pasaremos a obtener el volcado de los hashes de las contraseñas. Para ello utilizaremos el programa `pwdump6`. Para ello copiaremos en la raíz de la unidad C tanto el binario `pwdump.exe` como la librería `LsaExt.dll` y el servicio:

```
C:\>PwDump.exe 127.0.0.1 | findstr wadalberto

pwdump6 Version 1.0 by fizzgig and the mighty group at foofus.net
Copyright 2005 foofus.net

This program is free software under the GNU General Public License Version 2
(GNU GPL), you can redistribute it and/or modify it under the terms of the
GNU GPL, as published by the Free Software Foundation. NO WARRANTY, EXPRESSED
OR IMPLIED, IS GRANTED WITH THIS PROGRAM. Please see the COPYING file
included with this program and the GNU GPL for further details.

wadalberto1:1005:C1AD280AF36D87E4C623D0E1A0B0936D:61E93A89F1D140440F227D1EA0D
38DE6:::
wadalberto2:1006:C0597095BBBA4445AAD3B435B51404EE:6D557BA9B6A2BB193FEACCA3F73
2A809:::
wadalberto3:1007:43C4FCAC46A6159BAAD3B435B51404EE:CECD722E8E59FD0683236F814F2
61F4B:::
```

He filtrado la salida mediante el comando `findstr` para mostrar únicamente la información de las cuentas utilizadas para nuestro experimento.

El formato utilizado en la salida es el siguiente:

```
usuario:RID:HashLM:HashNT:::
```

Por lo que, y dado que estamos estudiando los hashes LM, deberemos fijarnos en la cadena mostrada en la tercera columna utilizando como separador el signo ':'. Además dividiremos la cadena en dos mitades de 16 bits cada una:

```
wadalberto1 C1AD280AF36D87E4 C623D0E1A0B0936D
wadalberto2 C0597095BBBA4445 AAD3B435B51404EE
wadalberto3 43C4FCAC46A6159B AAD3B435B51404EE
```

No es difícil observar como la segunda mitad del hash de la contraseña para los dos últimos usuarios es equivalente, y esto es debido a que ambos tienen una contraseña de menos de 8 caracteres. ¡Y esto será igual para cualquier usuario independientemente del valor asociado a su contraseña!

## 6. Como impedir el almacenamiento del hash LanMan

Ahora que ya conocemos la debilidad de los hashes LanMan seguro que nos gustaría encontrar una forma para que estos hashes no se almacenen. Pues la buena noticia es que este método existe, y la mala que aplicándolo perderemos la compatibilidad con sistemas Windows 9x/Me.

Desde el Service Pack 2 para Windows 2000 puede impedirse el almacenamiento del hash LanMan de las contraseñas de los usuarios, además de permitir a partir del momento de su implantación la creación de contraseñas de 15 caracteres o más.

Para ello deberemos buscar en el registro la siguiente clave:

```
HKLM\SYSTEM\CurrentControlSet\Control\Lsa
```

Allí encontraremos el valor `DWORD nolmhash`, establecido a 0 por defecto. Para activarlo solo tendremos que cambiarlo a 1 y volver a generar las contraseñas de todos los usuarios para que la modificación tenga efecto.

## 7. Referencias

Hackers en Windows 2000  
Editorial: McGraw-Hill

[Well-known security identifiers in Windows operating systems](#)

[How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases](#)

[How to use the SysKey utility to secure the Windows Security Accounts Manager database](#)

[Herramientas user2sid y sid2user](#)

[Windows NT/2000/XP/2003 password crackers - recovery, auditing, and PWDUMP tools](#)

[An experiment with Lepton's Crack - LM password cracking refinement](#)