

# Passwords: esos pequeños grandes amigos

Por Abraham Pasamar  
Consultor de Seguridad Informática  
esCERT-UPC / InetSecur

## Índice

1.	Introducción .....	2
2.	Necesidad del uso de passwords .....	2
3.	El valor de los passwords .....	3
4.	Desvelar los secretos es inherente a la a naturaleza humana.....	4
5.	Averiguando passwords: ataques por diccionario y fuerza bruta .....	4
6.	Crackeando passwords en Windows y Linux .....	5
7.	Tiempos para crackear passwords .....	8
8.	Como crear passwords "fuertes" .....	9
9.	Nociones sobre "autenticación fuerte" .....	11
10.	Las funciones de Hash.....	11
11.	¿qué es SHA-1?.....	12
12.	Glosario de términos y referencias.....	12

# 1.Introducción

Password [Inglés] = Contraseña [Castellano]

1. f. *Seña secreta que permite el acceso a algo, a alguien o a un grupo de personas antes inaccesible.*
2. f. *Segunda marca en animales y cosas para distinguirlos mejor.*
3. f. *Mil. Palabra o signo que, juntamente con el santo y seña, asegura el mutuo reconocimiento de personas, rondas y centinelas.*

Las contraseñas o "palabras secretas" han sido utilizadas a lo largo de la historia por las más diversas civilizaciones. Han sido protagonistas de guerras, conflictos, traiciones, etc. Pero siempre han perseguido un objetivo común, proteger secretos. En este artículo, nos centraremos concretamente en el uso de los passwords en medios computacionales (ordenadores, agendas personales, etc).

El primer uso de passwords en este tipo de entorno se remonta a mediados de los años 60. Ese fue el instante en el que alguien accedió a una computadora empleando una combinación de login<sup>1</sup> y password. Desde entonces, el uso de passwords para acceder a los sistemas no ha dejado de crecer, aunque en esencia, el sistema login/password ha cambiado bien poco desde 1960. Sin embargo, desde el punto de vista de la seguridad informática, muchas cosas han cambiado desde entonces. En aquel año, las personas que habían visto alguna vez un ordenador, no llenarían una sala de cine. Sin embargo, hoy en día, la mayoría de los habitantes del "primer mundo" no concebimos nuestra vida sin los ordenadores. Por tanto, cuando a alguien se le ocurrió la idea de utilizar un password en 1960, no debía de temer demasiado que alguien pudiese utilizar fraudulentamente dicho password, ya que solamente había unos cientos de personas en el mundo capaces de entender para que pudiera servir. Hoy en día utilizamos passwords para acceder al ordenador personal, al del instituto, al de la oficina, al cajero automático (un PIN es un tipo de password), al teléfono móvil, a nuestra cuenta de correo electrónico, a nuestro banco por Internet, etc. Por tanto, a priori, parece evidente que con el uso de passwords tratamos de limitar el acceso de otras personas a algo que nos pertenece. Es evidente que el uso de passwords se hace hoy en día inevitable y por tanto, deberíamos preguntarnos si las passwords que utilizamos son realmente necesarias, si son poco o muy importantes, si unas lo son más que otras y sobre todo, si son o no seguras.

## 2.Necesidad del uso de passwords

Si todos cerramos la puerta de nuestra casa con llave, incluso tenemos o hemos considerado el uso de un sistema de alarma, si cerramos el coche hasta cuando lo aparcamos en el garaje, si cerramos con llave el cajón de la oficina, ¿por qué se les da tan poco valor a los passwords? Probablemente, el problema es que los passwords son tan sólo una palabra en lugar de una flamante cerradura metálica, o quizá la mayoría de los usuarios no son conscientes de qué o de quién hay que protegerse con el uso de los passwords.

Estamos pasando a una velocidad vertiginosa de la "era papel" a la "era digital". Prácticamente en cualquier acto de nuestras vidas aparece algún tipo de registro digital. Al pagar con las tarjetas del banco, al fichar con la tarjeta de la empresa, al crear una hoja de cálculo con los gastos de nuestro hogar, al introducir un e-mail o un nuevo contacto en nuestra agenda del programa de correo electrónico, etc.

Estos datos, contienen en muchos casos información personal o de carácter confidencial que deben ser protegidos, pues mucha gente podría sacar partido de ellos. Debemos empezar a considerar la información digital como un bien muy preciado, tal y como lo hacemos con sus actuales equivalentes en papel o con nuestros bienes personales.

Los passwords nos permiten proteger la información. Su objetivo es por tanto **autenticar**<sup>2</sup> a alguien o a algo para que no acceda a determinado lugar o a determinada información. En otras palabras, lo que se busca mediante el uso de un password es proteger algo mediante la comprobación de la identidad de quien posea la palabra secreta.

### 3.El valor de los passwords

Con el término valor quiero referirme a la importancia de los passwords en un contexto determinado. Si por ejemplo, alguien descubre mi password para entrar a mi suscripción para leer una revista on-line, no es para mi tan importante como si alguien descubre mi password para acceder a mi cuenta bancaria o a mi historial clínico o a mi correo electrónico. Pero esta importancia es relativa y personal, y cada uno de nosotros debería ponderar que es más importante para él en cada caso.

Desgraciadamente, estamos tan acostumbrados a introducir passwords en los sistemas informáticos, que no siempre nos detenemos a evaluar importancia de ese password y los conceptos de seguridad que hay implicados. Si tenemos 5, 10, incluso 20 lugares diferentes en los que utilizamos un password, puede que hayamos puesto el mismo password para acceder a todos estos lugares. Si en alguno de ellos hay un punto débil y alguien descubre el password, podrá tener acceso a todos los demás sitios en los que tengamos ese password. Si por el contrario somos conscientes de este problema y utilizamos diferentes passwords, puede ocurrir que estos sean todos triviales y fáciles de obtener o puede que sean todos muy buenos e imposibles de recordar, lo cual nos llevaría a tener la tentación de apuntarlos en un papel o en un post-it sobre la pantalla del ordenador: grave error, si tenemos en cuenta que una de las mejores fuentes para conseguir passwords es precisamente buscar en las agendas personales, los papeles del cajón o mirar los pos-it que hay pegados en los monitores o sobre las mesas.

Lo primero que deberíais saber es que la mayoría de los passwords que utiliza la gente son inseguros. ¿Por hago esta afirmación tan rotunda? Porque en general, se utilizan como passwords palabras que se puede recordar fácilmente, es decir palabras comunes, como las que hay en el diccionario. El segundo error, es que además suelen ser palabras que están de algún modo relacionadas con cada persona. Es decir, que son el nombre de su mascota, de su mujer o de su novia. A pesar de que muchos combinan estas palabras con su año de nacimiento o con los últimos dígitos del año actual, o del año en que se casó, deberían saber que esto es absolutamente insuficiente. Si nuestro password está formado una combinación de palabras o datos que estén en un diccionario, por números o por una combinación de palabras y números, éste será un password inseguro, un password débil. Si además, es predecible porque contiene información relacionada con nuestro entorno o persona, será un password extremadamente débil.

Considero por tanto, que es muy bueno hacer una reflexión de cuanto tiempo y esfuerzo dedicar a "crear" un nuevo password, o a reemplazar algunos de los que ya tenemos, teniendo siempre presente que nivel de seguridad requiere la información que queremos proteger. Al final del artículo se explican algunas técnicas para crear passwords fuertes.

## 4. Desvelar los secretos es inherente a la naturaleza humana

A mucha gente le parecerá una tontería y una pérdida de tiempo el hecho de tratar de averiguar las passwords de otras personas. Pero seguro que algunos de vosotros estaréis pensando que os gustaría saber el password de algún compañero o amigo y poder acceder a sus secretos, a ese rincón de su intimidad tan celosamente guardado. Algunos lo querrían sólo por curiosidad, para saber que password han utilizado y conocer así un poco más la personalidad de su "amigo". A otros quizá les mueva la curiosidad por saber que fotografías guardará en su disco duro o que clase de e-mails envía o que páginas web visita. Y algún otro, seguro que va más lejos y piensa en sacar un beneficio más suculento de esa información. Por tanto, no es nada extraño pensar que puede haber mucha gente dispuesta a averiguar nuestro password e invadir nuestra intimidad.

La mayoría de los que intenten averiguar un password recurrirán a las técnicas que se explicarán en el siguiente punto. Utilizarán determinados programas para tratar de averiguar passwords comunes basados en palabras que están el diccionario, combinaciones de números, etc. En la medida en que nuestro password sea más fuerte, más difícil se lo pondremos y puede que desistan en su intento. Los passwords sencillos, serán con toda seguridad averiguados en un plazo muy corto de tiempo y nuestros secretos quedarán al descubierto.

## 5. Averiguando passwords: ataques por diccionario y fuerza bruta

Como hemos comentado, existen programas informáticos cuya finalidad es averiguar passwords. Y hay para todos los gustos, para averiguar el password con el que se ha protegido una hoja de cálculo, un documento de texto, una cuenta de usuario, un fichero comprimido, una cuenta de correo electrónico, etc. Para lograrlo, se utilizan diferentes técnicas, entre las que destacan las siguientes:

- **Ataque por diccionario**
- **Ataques híbridos**
- **Ataques por fuerza bruta**

Todos estos tipos de ataque tienen un patrón común que consiste en "probar", uno por uno, los diferentes passwords de un listado contra un objetivo.

- **Ataque por diccionario:**

En este tipo de ataque consiste en que, se selecciona en primer lugar un listado de palabras comunes, éste listado puede ser venir junto con el programa o podemos proporcionárselo nosotros (probad a buscar en Google<sup>3</sup> ficheros de diccionarios y veréis lo que encontráis). En segundo lugar, se prueban 'contra' el documento, la aplicación o determinados ficheros que almacenan passwords, todas y cada una de las palabras del diccionario proporcionado. Como podéis comprender, el tiempo que tardará el programa depende de varios factores. Por ejemplo, será más rápido 'atacar' un fichero de passwords que se encuentra en nuestro PC localmente, que uno que se encuentra en un ordenador de la red de área local (LAN<sup>4</sup>), y será todavía más lento realizar este ataque a través de Internet, ya que en general la conexión será mucho más lenta.

Además el programa puede estar más o menos optimizado, de manera que sea capaz de probar más o menos passwords en un tiempo determinado. Por tanto, dos

programas que traten de averiguar los passwords de un fichero local de passwords empleando un diccionario exactamente igual en el mismo ordenador pueden tardar tiempos diferentes en función de como estén programados cada uno, aunque ambos funcionen correctamente y lleguen al mismo resultado.

Otro factor a tener en cuenta en el tiempo que tardará uno de estos programas en averiguar un password, es el tipo de hardware sobre el que se ejecuta, ya que a mayor velocidad de procesamiento, mayor capacidad de cálculo y por tanto más intentos por segundo podrá realizar. También puede haber una limitación por el tipo de red a través de la cual se accede o por el tipo de documento, protocolo de comunicación<sup>5</sup> o aplicación que se ataca.

- **Ataque híbrido:**

Este tipo de ataque utiliza las palabras del diccionario y las transforma de diversas formas para tratar de averiguar aquellas passwords que son combinación de una palabra común con algún número o carácter especial (cosa muy habitual). Las transformaciones más utilizadas por la gente son añadir números delante o detrás de las palabras o algún signo de puntuación. Otro caso habitual son las sustituciones 'leet speak', que consisten en una forma especial de escribir las palabras en las que algunas letras son sustituidas por dígitos que tienen cierto parecido gráfico. Por ejemplo, la A se sustituye por un 4 la E por un 3, la I por un 1 y la O por un 0. Existen algunas combinaciones más, pero básicamente estas son las más utilizadas. Mucha gente considera este tipo de transformación como un arma infalible para crear passwords seguros, pero la mayoría de los programas que realizan ataques híbridos ya implementan este tipo de combinaciones y acabarán obteniendo la password a pesar de que tardarán más tiempo en hacerlo que si la palabra está escrita en el lenguaje tradicional.

Este tipo de programas, permiten además que el usuario decida que tipo de combinaciones desea probar y cuales no para hacer un ataque más específico y selectivo.

- **Ataque por fuerza bruta:**

¿Y si los anteriores ataques no tienen éxito? Entonces se procede al ataque por fuerza bruta. Este tipo de ataque prueba un listado de passwords generado a partir de todas las combinaciones de letras [A-Z][a-z], números [0-9] y caracteres [!"\$%&/()\*+, etc.] posibles y por tanto es mucho más lento. Siempre resultará mucho más eficaz realizar un ataque por diccionario o híbrido antes de un ataque por fuerza bruta, ya que este último es mucho más lento debido a que el número de passwords que se prueban es muy superior al de los otros tipos ataques.

## **6. Crackeando passwords en Windows y Linux**

Existen, como ya hemos comentado, numerosos tipos de passwords que sirven para proteger ficheros (pdf, doc, xls, zip, etc.), para acceder a dispositivos (routers, switches, ROMs, PDAs, etc.), passwords de protocolos de comunicaciones (telnet, ssh, pop3, etc.) y password que sirven para identificar a un usuario que inicia una sesión en el sistema operativo de una computadora (Windows, Linux, Unix, AS/400, etc.). Sobre estos últimos, vamos a tratar más en profundidad en este punto, ya que son algunos de los passwords más utilizados y a la par más codiciados.

Existen diferencias entre como gestionan los passwords los dos sistemas operativos más comunes: Windows y Linux. No todas las versiones de Windows tratan los passwords de igual modo, pero existen similitudes entre Windows NT, Windows 2000, Windows XP y Windows 2003, así como entre Windows 95,98 y Me.

- **Windows**

En Windows, los passwords no se guardan directamente en un archivo como texto plano<sup>6</sup>, sino que se almacena el Hash<sup>7</sup> de cada password. La función de Hash, es un algoritmo matemático que transforma el password en una secuencia alfanumérica de longitud fija (Ver el punto "Las funciones de Hash"). Es una manera de transformar el password con el fin de protegerlo de los ojos curiosos. Existen diferentes funciones de Hash, pero para proteger los passwords Windows utiliza dos tipos **LM**<sup>8</sup> (Win95/98/Me) y **NTLM**<sup>9</sup> (WINNT/2000/XP/2003). Una password transformada mediante una función de Hash tendrá más o menos este aspecto "F923482E5BF859B28CD74AF4CB5D14CE". Pero, ¿donde se guardan los *hashes* de las passwords? pues se guardan en un archivo llamado **SAM**<sup>10</sup> (Security Accounts Management Database) que puede encontrarse en **%systemroot%\system32\config**. Donde %systemroot% es el directorio raíz donde se encuentran los archivos del sistema, que puede ser el directorio WINNT o WINDOWS según el caso. Puede que os estéis preguntando si es posible obtener este fichero y tratar de averiguar las passwords. Pues depende, si sois administradores<sup>11</sup> del equipo sí, de lo contrario, no podréis obtenerlo tan fácilmente. Hay que tener en cuenta que Windows NT en principio no protege el fichero de *hashes* de ser copiado y editado, pero si se activa el **syskey**<sup>12</sup>, sí que se protege el fichero mediante una función de Hash **MD5**<sup>13</sup>. En Windows 2000 está encriptación viene por defecto activada. Esto quiere decir que además de obtener el fichero que contiene los *hashes* de los passwords, debemos desencriptarlo, para poder acceder a los *hashes*.

En teoría, al estar protegido por una transformación MD5, la labor de 'atacar' el fichero SAM es a priori prácticamente imposible, pero existe un eslabón débil en el proceso en que Windows utiliza dicho fichero. Cuando introducimos nuestro password en el sistema, lo que hace Windows para comprobar que éste coincide con el password que está almacenado en la SAM es lo siguiente:

Transforma el password que introducimos mediante la función de Hash LM o NTLM, según el caso. Después accede al fichero SAM, lo desencripta utilizando un password que sólo Windows conoce y comprueba si el Hash calculado coincide con el Hash que hay en la SAM para dicho usuario. Si es correcto le da acceso al sistema y de lo contrario se lo deniega.

Es en la forma en la que se accede a los *hashes* en el momento de la comprobación, donde está el punto débil del proceso, y existen técnicas que permiten acceder al fichero de *hashes* que en ese momento está desencriptado en la memoria RAM del ordenador y copiarlo.

Para poder llevar a cabo esta técnica, existen unas herramientas llamadas **pwdump2** y **pwdump3** para Windows 2000, que son capaces de acceder al fichero desencriptado de passwords y saltarse la poderosa encriptación MD5 del fichero.

En cualquier caso, si intentáis acceder al archivo de passwords SAM para copiarlo directamente, veréis que éste no es accesible, ya que Windows lo bloquea para su uso y no permite copiarlo. Una forma para saltarse dicho bloqueo es arrancar con un sistema operativo diferente utilizando un CD de arranque o un disquete, que soporte el sistema de fichero **NTFS**<sup>14</sup>, el cual nos permitirá acceder al disco y copiar

el fichero. Otra forma es mediante el archivo **SAM.\_** que Windows crea como backup, pero deberéis tener cuidado porque puede que contenga passwords antiguas. Deberéis mirar las fechas del fichero para saber si es reciente.

Una vez obtenido el fichero de *hashes* mediante **pwdump**, podremos tratar de averiguar las password mediante el programa **L0phtCrack**. En su última versión conocido como LC4 de la firma @stake. Mediante LC4 podemos realizar los diferentes tipos de ataque antes descritos, como son el ataque por diccionario, el híbrido y por fuerza bruta.

En cuanto a Windows 95 y 98, hay que decir que no son sistemas operativos pensados para ser seguros. Por tanto la protección de claves no es su fuerte. Las passwords se guardan en archivos PWL (archivos no protegidos) y su encriptación (LM) es propietaria de Microsoft y muy débil. Existen numerosos programas capaces de desencriptarlas.

La mayor debilidad de los passwords en los sistemas Windows más modernos (Windows NT/2000/XP/2003) es que para poder mantener la compatibilidad con las versiones precedentes, en los archivos de passwords SAM, se guardan tanto los *hashes* LM como los NTLM. Como los passwords LM son mucho más débiles, es relativamente fácil encontrar el password correspondiente a un Hash LM y luego atacar con dicha información el password NTLM. Los passwords LM tienen limitaciones como que el número máximo de caracteres está limitado a 7. Por tanto, si se introduce un password mayor, el sistema lo parte en fragmentos de 7 caracteres. Además, los passwords LM sólo utilizan letras mayúsculas, de manera que el número de combinaciones es mucho más limitado. Supongamos el siguiente ejemplo de password: "Pr1v4d01985", a primera vista no está mal, dado que la longitud es 11 caracteres. Hay mayúsculas, minúsculas y números. Además se a utilizado 'leet speak' para ocultar una palabra del diccionario. Un sistema Windows NT/2000/XP/2003, el sistema almacenará el Hash LM y el NTLM. El problema es que antes de almacenar el password LM partirá la palabra en 2. Y almacenara 2 passwords, por un lado "Pr1v4d0" y por otro "1985". Además, transformará "Pr1v4d0" en "PR1V4D0" antes de calcular el Hass LM. Si nuestro programa de Crack soporta las combinaciones "leet speak" no tardará apenas tiempo en averiguar la primera parte del password "PR1V4D0" puesto que es una variación de una palabra del diccionario y además sólo ha de probar las letras mayúsculas. La segunda parte del password es todavía más sencilla de encontrar puesto que es un número de cuatro dígitos. Por tanto, en un tiempo muy breve habrá encontrado el password LM, combinación de las dos partes anteriores. A continuación, para averiguar el password NTLM, que dará acceso al sistema, sólo ha de probar las diferentes combinaciones de mayúsculas y minúsculas sobre el password LM encontrado "PR1V4D01985", de manera que llegar a obtener el verdadero password "Pr1v4d01985" no le llevará mucho tiempo. Como puede apreciarse, la el tener que mantener la compatibilidad de ambos sistemas de Hash, hace un flaco favor al sistema NTLM, que es mucho más robusto que el LM.

#### ○ **Linux**

En el sistema operativo linux, la gestión de passwords es bastante diferente. En los sistemas **Unix**<sup>15</sup> y **Linux** los passwords se han guardado tradicionalmente en el fichero /etc/password. Esto en sí constituía un problema de seguridad, ya que este fichero tiene permisos de lectura para todos los usuarios y por tanto puede ser copiado para ser tratado por un programa de cracking de passwords.

Debido a este problema, se creo un segundo fichero para "esconder" las passwords y se separo del fichero principal. Este nuevo fichero es /etc/shadow y contiene las passwords y algunos datos del usuario. Por otra parte el fichero /etc/passwd ya no contiene las passwords, sino que contiene el resto de los datos de los usuarios. El fichero /etc/passwd sigue siendo de lectura para todos los usuarios, mientras que el fichero /etc/shadow sólo es accesible por 'root'<sup>16</sup>. En cualquier caso, a día de hoy hay distribuciones de linux que no implementan directamente las shadow passwords por defecto, aunque suelen ofrecer la posibilidad de hacerlo durante la instalación. Para saber si estamos ante un sistema que utiliza el fichero /etc/passwd para guardar las passwords teclearemos en una shell<sup>17</sup>:

```
$ cat /etc/passwd
```

y si la salida es algo similar a esto:

```
root:JI$!4P1gKLxVM3:0:0:root:/root:/bin/bash
```

quiere decir que los password se guardan en el fichero /etc/passwd. Si en lugar de aparecer en el segundo campo (los campos se separan mediante el caracter ':') la clave encriptada, apareciese una secuencia 'x' querría decir que los passwords se guardan en el fichero /etc/shadow.

```
cat /etc/passwd:
```

```
root:x:0:0:root:/root:/bin/bash
```

Linux no encripta el fichero de passwords, simplemente le da permisos de lectura únicamente al 'root' pero no encripta el fichero en sí. Lo que sí hace es encriptar cada una de las passwords, bien mediante la función crypt()<sup>18</sup> o mediante el algoritmo Md5. Crypt() es una implementación derivada del algoritmo de encriptación DES. Actualmente se utiliza crypt(3), que es la correspondiente variante del 3DES, ya que DES no se considerará seguro hoy en día. Pero desde hace tiempo, se implementa también Md5. Por tanto, cada password de cada usuario de linux, está encriptada con 3DES o MD5 a diferencia de Windows donde cada password está transformada mediante una función de Hash propietaria llamada LM o NTLM (WinNT/XP/2000).

El programas más utilizado en el entorno Linux, para atacar un fichero de passwords /etc/shadow al que se ha tenido acceso, es utilizando el programa **John the Ripper** (Juanito el Ripeador).

## 7. Tiempos para crackear passwords

Ya hemos visto que es más práctico realizar ataques por diccionario e híbridos que ataques por fuerza bruta. En cualquier caso, todas estas pruebas llevan tiempo. Para estimar el tiempo que puede tardar un ordenador en realizar un ataque, pondremos un ejemplo basado en un procesador actual a 1 GHz (dentro de unos años alguien se reirá mucho si lee esto, pero ...). Éste es capaz de realizar aproximadamente 1 billón (1.000.000.000 operaciones por segundo). Esto es un poco exagerado, pero sirve como ejemplo. Supongamos que hay un programa que calcula la función de Hash de las diferentes combinaciones passwords y la compara con las *hashes* del fichero que queremos crackear. Supongamos que para cada una de las passwords el programa necesita, por ejemplo, 1000 instrucciones del procesador. Entonces 1 billón intrucciones/s entre 1000 instrucciones/password, nos da 1.000.000 de passwords/segundo. Es decir, un procesador a un GHz puede

probar aproximadamente 1.000.0000 de passwords/segundo. Esto parecen muchos intentos por segundo, pero, ¿cuanto tardará en averiguar passwords que estén el diccionario, y variantes de estos passwords y passwords formados por letras mayúsculas, minúsculas, números y otros caracteres?

Para verlo más claro veamos el siguiente cuadro:

Tipo de Passwords	Número de combinaciones	tiempo máximo de crackeo
palabra del diccionario	1.000.000	1 segundo
variantes sin 'leet speak', Mayusculas, Reverse	5.000.000	5 segundos
[a-z] minusculas hasta 4 caract.	456.976 (26 <sup>4</sup> )	aprox. 0,5 segundos
[a-z] minusculas + [0-9] hasta hasta 6 caract.	2.176.782.336 (36 <sup>6</sup> )	2.176 segundos (36 minutos)
[a-z] + [A-Z] + [0-9] hasta 6 caracteres	56.800.235.584 (56 <sup>6</sup> )	56.800 segundos (16 horas)
[a-z] + [A-Z] + [0-9] + [!"\$%&/()*], etc hasta 6 caract.	208.422.380.089 (77 <sup>6</sup> )	208.422 segundos (2,4 días)
[a-z] + [A-Z] + [0-9] + [!"\$%&/()*], etc hasta 8 caract.	1.235.736.291.547.681 (77 <sup>8</sup> )	1.235.736.292 segundos = 39 años !!!

Es evidente que introduciendo una cierta complejidad en el password, añadiendo mayúsculas y minúsculas, números y algún carácter poco común, los tiempos para averiguarlo se disparan.

## 8. Como crear passwords "fuertes"

Quizá para nuestro amigo de 1965 que introdujo su primer password en una computadora de la época no era una mala idea el utilizar un password como "mom65", por ejemplo, pero hoy en día el estado de la técnica permite que ese password "caiga" en cuestión de segundos, minutos como mucho. Además la escalada de velocidad de los procesadores es exponencial, por tanto en pocos años no servirán las técnicas que estamos aprendiendo ahora.

¿como debe ser por tanto un passwords seguro? aquí es donde pueden ayudarnos un poco las matemáticas. Si creamos un password utilizando sólo las letras minúsculas, tendremos por tanto 26 caracteres diferentes entre la a y la z. ¿Cuantos passwords posibles hay si la longitud del password es de 4 caracteres? la respuesta es  $26^4 = 456976$  passwords. Si os han parecido muchos, debéis tener en cuenta que un ordenador actual podría probar esa cantidad de passwords en cuestión de segundos :( ¿y si el password fuera de 8 dígitos? tendríamos  $26^8 = 208.827.064.576$  nada menos que doscientos ocho mil ochocientos veintisiete millones de passwords. Parece que es un número mucho mayor, pero os aseguro que puede caer en cuestión de horas. Parece por tanto razonable buscar

un password que requiera más tiempo para ser descubierto, ¿pero cuanto? Pues depende, porque algunos programas informáticos limitan el número máximo de dígitos o caracteres que pueden usarse y por tanto nos condicionan la seguridad del mismo desde su propia creación. Pero como consenso se ha establecido que a día de hoy un password seguro o "fuerte" es el que tiene una longitud entre 6 y 8 caracteres y es una combinación de letras minúsculas mayúsculas, dígitos numéricos y caracteres especiales y es además "imposible" de recordar. Por tanto no debe de tener significado ninguno para una persona que lo viese escrito (y, por supuesto, no debe escribirse nunca en ningún papel, post-it, agenda o similar). Supongamos es siguiente passwords "fuerte":

Mpg10Md\$

aparentemente no tiene ningún significado y realmente parece "imposible" de recordar. Pero hay técnicas que permiten que creamos y recordemos passwords como este con facilidad. Por ejemplo el password anterior podría haberse creado del siguiente modo:

Pensemos en una frase, por ejemplo: "Mi padre gana diez Millones de Dolares".

Ahora escribamos las primeras letras de cada palabra y transformemos el número "diez" en un "10" numérico. Además sustituiremos la palabra "dolares" por el símbolo \$. Así pues nos quedará:

Mpg10Md\$

y cada vez que tengamos que introducirla pensaremos en esta frase y nos acordaremos del password sin problemas.

En definitiva, este método de crear passwords se basa en acordarse de una frase. La frase puede ser sencilla, la cuestión es que tenga un número mínimo de palabras que nos permitan crear un password suficientemente largo, tomando la primera letra de cada palabra. Por supuesto otra forma es tomar dos o tres letras de cada palabra. Cada uno es libre de crear su propio método. Lo importante es que no se nos olvide la frase y que las reglas de sustitución que utilicemos no sean demasiado complejas como para que lleguemos a olvidarlas.

Otros ejemplos podrían ser:

100*100pE	"Cien por cien pura Energía"
imetdJL	"Admiro mucho el trabajo de José Luis"
L2usA**	"Los dos últimos son Asteriscos"
EuldIM...	"En un lugar de la Mancha ..."
2+2s4!	"Dos y dos son cuatro!"

etc...

Por tanto y recapitulando, veamos que cosas NO deben hacerse a la hora de crear o trabajar con passwords:

- Apuntarlo, pegarlo con un post-it sobre el monitor o decirse a alguien
- Utilizar palabras que estén en el diccionario
- Las sustituciones 'leet speak' son insuficientes
- concatenar palabras es mejor que palabras únicas pero no es suficiente (ej: "perro+gato")
- utilizar passwords menores de 6 caracteres

Veamos lo que SÍ se debe hacer para crear un buen password:

- Utilizar passwords de al menos 6 caracteres
- Que sean "imposibles" de recordar
- Que contengan letras mayúsculas [A-Z] y minúsculas [a-z], dígitos numéricos [0-9] y caracteres especiales p.ej: [!"·#@\$\$%&/()='^[ ]{}' \_.,:;"]

## 9. Nociones sobre "autenticación fuerte"

Reflexionemos un momento sobre el concepto de autenticación. Veámoslo con un ejemplo: si yo os entrego mi tarjeta de visita en la que pone que soy Cill Hates, presidente de Dicrosoft Corporation y vosotros no me conocéis de nada, pensaréis que soy realmente quien digo ser. En realidad, una tarjeta de visita es un modo de autenticación, débil, pero es un método. Si a continuación, os entrego mi DNI con ese nombre y mi foto, probablemente estaréis mucho más seguros de mi identidad, porque el DNI es un buen método de autenticación, aunque no infalible. Si tomamos mis huellas digitales y éstas se contrastan con el DNI y coinciden, habrá pocas dudas sobre mi identidad. Estamos en este caso ante una autenticación muy fuerte.

Una definición sencilla de autenticación sería decir que "Autenticar es probar la identidad de alguien o algo".

Para probar la identidad de alguien, existen tres métodos:

- Conocer algo (p.ej. un password)
- Poseer algo (p.ej. un DNI)
- Ser algo (p. ej. una identificación mediante "parámetros biométricos: huellas, retina, iris, voz, etc.")

En general, es más seguro un método basado en "ser algo" que uno basado en "poseer algo" que uno basado en "conocer algo". Pero una cosa es que sea más seguro y otra muy distinta es que sea sencillo, funcional y asequible.

Cuando utilicemos, al menos, dos de ellos conjuntamente, estaremos hablando de "autenticación fuerte".

Existen hoy en día, sistemas que utilizan autenticación fuerte, estos son por ejemplo los "tokens" o sistemas de autenticación biométricos. Pero no siempre es posible utilizarlos ya sea por razones de complejidad o económicas y por tanto, los passwords tradicionales continúan siendo hoy en día el método más popular para la autenticación.

## 10. Las funciones de Hash

Entre las funciones de cifrado, existen unas de enorme utilidad llamadas "Funciones de Hash". "Funciones Resumen" o de "integridad". Estas curiosas funciones, son una pieza fundamental en las aplicaciones de Internet que requieren protección de la información, pues permiten autenticar un mensaje y garantizar su integridad. Es decir, mediante el uso de estas funciones, es posible conocer si un mensaje ha sido o no alterado durante su envío a través de Internet o su almacenamiento.

Podemos imaginarnos una Función de Hash como una caja negra con una entrada y una salida. Por la entrada, introducimos un mensaje de longitud variable y por la salida obtenemos un código (hash) de longitud fija, equivalente a un número de unos 50 dígitos decimales. Dos mensajes diferentes generan "hashes" diferentes, a priori, pero accidentalmente dos mensajes pueden tener el mismo hash, con una probabilidad de 10-50, pero lo que debe garantizar el algoritmo es la imposibilidad de encontrar un segundo mensaje con igual hash que otro. Además, dado un mensaje, es fácil y rápido calcular su "hash", pero es imposible reconstruir el mensaje original a partir del código resumen (hash).

#### **Resumiendo:**

- Todos los hashes generados con una función de hash tienen el mismo tamaño, sea cual sea el mensaje utilizado como entrada.
- Dado un mensaje, es fácil y rápido mediante un ordenador calcular su hash.
- Es imposible reconstruir el mensaje original a partir de su hash.
- Es imposible generar un mensaje con un hash determinado.
- Es casi imposible (10-50) que dos mensajes diferentes tengan el mismo código hash.

## **11. ¿qué es SHA-1?**

SHA-1 es una función matemática para calcular un código "resumen" de un mensaje o documento electrónico de 160 bits. Este código es el que se usa para proteger los ficheros contra modificaciones no autorizadas (preservar su integridad), permitiendo la detección de troyanos en programas de ordenador, o evitando que los virus modifiquen los listados de firmas de los antivirus para evitar ser detectados.

Pero sobre todo, éste es el algoritmo empleado para evitar la suplantación de servidores web seguros, empleados para servicios de comercio electrónico, financieros, gobierno electrónico, y también para firmar electrónicamente documentos y certificados de identidad electrónica, como el DNI electrónico que está a punto de lanzar el gobierno español, o los que emiten el ministerio de hacienda o los gobiernos autonómicos de Catalunya, Valencia y Euskadi.

## **12. Glosario de términos y referencias**

---

<sup>1</sup> **Login:** nombre de usuario para acceder a un determinado sistema informático. Generalmente se acompaña de un password, ya que el login puede ser público y el password es secreto.

<sup>2</sup> **Autenticar:** validar la identidad de alguien o algo

<sup>3</sup> **Google:** el buscador de información más utilizado de Internet, cuya dirección es <http://www.google.com>

<sup>4</sup> **LAN:** Local Area Network (red de área local)

<sup>5</sup> **Protocolo de comunicación:** Existen diferentes métodos de comunicación entre los ordenadores a través de una red. La forma en la que se definen dichos métodos de comunicación se denominan protocolos. Algunos de los más famosos, que forman parte de la colección de protocolos de Internet (TCP/IP), son: TELNET, POP, SMTP, DNS, etc.

- 
- <sup>6</sup> **Texto plano:** Un fichero en texto plano es aquel que almacena los datos sinningún tipo de formato, es decir, únicamente almacena texto y debe poder leerse con el más simple de los editores de ficheros.
- <sup>7</sup> **Hash:** Una función de Hash, es una función matemática para calcular un código "resumen" (*hash*), de un mensaje o documento electrónico. Las características principales de una función de Hash son:
- Todos los *hashes* tienen el mismo tamaño independientemente del tamaño del mensaje original
  - Es fácil y rápido calcular una función de Hash con el uso de un ordenador
  - Es "imposible" reconstruir el mensaje original a partir de su Hash
  - Es "imposible" generar un mensaje para obtener un Hash determinado
- <sup>8</sup> **LM:** Lan Manager, función hash para proteger passwords usada por Windows 95,98 y Me
- <sup>9</sup> **NTLM:** NT Lan Manager es la versión para NT del LAN Manager
- <sup>10</sup> **SAM:** Security Accounts Management Database (Base de datos de gestión de las cuentas de seguridad)
- <sup>11</sup> **Administrador:** usuario con privilegios de administración sobre un sistema. El usuario Administrador tiene control total sobre el sistema que administra.
- <sup>12</sup> **Syskey:** Es una herramienta para proteger el acceso a la base de datos **SAM** donde están almacenados los *hashes* de los passwords de los usuarios.
- <sup>13</sup> **MD5:** es una función hash que transforma una cadena de datos de entreda de cualquier longitud en otra cadena más corta de longitud fija (128 bits ó 16 caracteres). Sirve para verifica la integridad de los datos y comprobar que no han sufrido alteraciones o manipulaciones. Actualmente ya no se considera segura.
- <sup>14</sup> **NTFS:** (New Technology File System) Es un sistema de archivos diseñado específicamente para Windows NT, con el objetivo de crear un sistema de archivos eficiente, robusto y con seguridad incorporada desde su base.
- <sup>15</sup> **Unix:** Es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.
- <sup>16</sup> **root:** En los sistemas tipo UNIX, root es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multi usuario). root es también llamado superusuario.
- <sup>17</sup> **shell:** Es una parte fundamental del sistema operativo, encargada de ejecutar las órdenes básicas para el manejo del sistema. Suelen incorporar características tales como control de procesos, redirección de entrada/salida y un lenguaje de órdenes para escribir programas por lotes o scripts.
- <sup>18</sup> **Crypth ( ):** Es una función hash que transforma una cadena de datos de entreda de cualquier longitud en otra cadena más corta de longitud fija. Usada antiguamente en sistemas Unix. Evoluciono a la función crypth(3). Actualmente no se considera segura.