Universidad Politécnica de Cartagena

Escuela Técnica Superior de Ingeniería de Telecomunicación

SEGURIDAD EN REDES

Práctica 2: Seguridad en el Sistema Operativo (II)

Funciones *hash* aplicadas a la autenticación de usuarios

Autores: Josemaría Malgosa Sanahuja
María Dolores Cano Baños

SEGURIDAD EN REDES

Funciones *hash* aplicadas a la autenticación de usuarios PRÁCTICA 2 (segunda parte)

El uso más frecuente de funciones *hash* es para firmar documentos y para autenticar usuarios. En esta práctica se ejemplificará el uso que hace el sistema operativo Linux de los *hash* DES, MD5, SHA-256 y SHA-512 para autenticar usuarios. En otra práctica se estudiará el uso de los *hash* para firmar documentos.

En Linux, el método de autenticación más común es mediante un *login* y un *password*. El sistema comprueba si el usuario existe buscando el *login* en el fichero */etc/passwd*. En caso de existir, calcula el *hash* del *password* introducido por el usuario y lo compara con el almacenado en el fichero */etc/shadow*. Solo en el caso de que coincidan, el usuario accede a su cuenta de Linux.

Estos sistemas sufren dos tipos de ataques:

- Ataque por fuerza bruta: Dado un alfabeto (por ejemplo [0-9A-Za-z] de 62 caracteres) y fijando la longitud de la contraseña a L caracteres, se trata de calcular el *hash* de todas los posibles combinaciones de tamaño L hasta encontrar una coincidencia con el *hash* almacenado en la base de datos (en nuestro caso, en */etc/shadow*)

- Ataque por diccionario: Existe una base de datos que almacena multitud de contraseñas utilizadas por usuarios reales, así como sus posibles variantes. Se trata de calcular el *hash* de cada una de ellas hasta encontrar una coincidencia con el *hash* almacenado en la base de datos (en nuestro caso, en */etc/shadow*)

1. ¿Cuál es la estructura del fichero */etc/passwd*? ¿Cuáles son los permisos del fichero */etc/passwd*?

2. ¿Cuál es la estructura del fichero */etc/shadow*? ¿Quién tiene acceso a este fichero? ¿Por qué no se acostumbra a almacenarse los *hash* de las contraseñas en */etc/passwd*?

Los algoritmos se utilizan para el cálculo del *hash* son DES, MD5, SHA-256 y SHA-512 (el DES es el realidad algoritmos de cifrado adaptados para que funcionen como *hash*). Todos estos algoritmos admiten SALT y ROUND.

3. ¿Qué es un SALT y un ROUND? ¿Qué algoritmos admiten ROUND? ¿Cuál es el formato con el que se almacenan los *hash* en el archivo */etc/shadow*?

El programa *makepasswd.c* (ver anexo) calcula el *hash* de la contraseña que se le proporcione. Estudiar el código de dicho programa y responder razonadamente a las siguientes preguntas:

4. ¿Qué opciones admite el programa? ¿Es necesario que el usuario especifique un SALT?

5. ¿Qué utilidad tienen los SALT y ROUND? Si se tiene acceso al archivo */etc/shadow*, ¿proporcionan los SALTS y ROUNDS mayor seguridad?

6. ¿Cuál debería ser el valor del *hash* del usuario *alumno* en el fichero */etc/shadow* si quisiéramos que la contraseña fuera lssei? ¿Qué hace el comando *passwd*?

7. ¿Cree que es factible disponer del fichero */etc/shadow* de un ordenador del cuál usted no tiene permisos de super-usuario? ¿Cree que es factible saber la longitud aproximada de una contraseña?

1. -rw-r--r-- 1 root root 2008 Oct 19 19:24 passwd

```
<nombre>,<password><uid>, <gid>,<descripción opcional> <carpeta>, <shell>
```

<nombre> No se admiten números al comienzo de un nombre de usuario.

<password> Una «**x**» indica que el password está almacenado en /etc/shadow, en el caso de ser una «**!**» es que el usuario está bloqueado. Si tiene «**!!**» es que no tiene.

<uid> Cada usuario lleva un no identificador (uid) entre 0(root) y 65535. Se reservan algunos para usuario root(el cero siempre), y para usuarios de servicios varios del sistema.

Red hat y derivados entre 1 y 499.
Debian y derivados entre 1 y 999.

<gid> – grupo id, cada usuario tiene un id de grupo principal, pero puede pertenecer a más grupos.

<carpeta> La usará como la carpeta de inicio del usuario, al iniciar sesión con él será la que cargue por defecto.

<shell> Los usuarios de servicios y usuarios con permisos limitados no deben tener shell, es decir iniciar sesión en consola, normalmente se les deja con /usr/bin/nologin o /bin/false

2. -rw-r----- 1 root shadow 1294 Oct 19 19:24 shadow

```
<nombre><password cifrado><1><2><3><4><5><6>
```

<nombre> Nombre del usuario.

<password cifrado> Pues eso…

1-Días transcurridos desde 1-1-1970 donde el password fue cambiado por última vez.
2-El mínimo número de días entre cambios de contraseña.
3-Días máximos de validez de la cuenta.
4-Días que avisa antes de caducar la contraseña.
5- Días después de que un password caduque para deshabiltar la cuenta
6- Fecha de caducidad. días desde 1-1-1970, donde la cuenta es deshabilatada y el usuario no podrá iniciar sesión

Las contraseñas cifradas no se almacenan en *passwd*, ya que este archivo puede ser leído por cualquier usuario y con programas de descifrado se pueden descubrir

3. El formato del hash en *shadow* es con ROUND (depende del algoritmo de cifrado).

- SALT: Es un número de dígitos aleatorios que se le agrega a la contraseña ya sea al principio o al final y que el usuario no conocerá. Con esto, la contraseña se hace de mayor longitud y por lo tanto más compleja, de esta manera será mucho más difícil encontrar el hash en una tabla y más aún obtener la contraseña, ya que está combinada con el salt.
- ROUND: Consiste en repetir el SALT n veces.

4. No es necesario que el usuario especifique un SALT.

```
localhost:~/Downloads # ./makepasswd
Password must be specified

Usage: makepasswd [OPTIONS]... password
Crypts the PASSWORD using crypt(3).

-m, --method=TYPE     select method TYPE
-s, --salt=SALT       use the specified SALT
-r, --rounds=NUMBER   use the specified NUMBER of rounds
-h, --help            display this help and exit
-v, --version         output version information and exit
```

5. Sí
6. El comando *passwd* sirve para cambiar la contraseña del usuario.

```
localhost:~/Downloads # ./makepasswd -m sha-512 lssei
$6$Qb6xy05/qtSRj$p5Xo2LBu4bQ2A/PTEmgCHloIxk7vLcZhu8dJRwtizdFJP4AL.WANDKdZW/ukwCWr9zR15VCIFo4VXh4DfRU/N/
```

7. No se puede estimar la longitud de una contraseña. Ni tener el archivo /etc/shadow sin ser super-usuario.
8. La contraseña es "*paco*"

```
localhost:~/Downloads # time ./crack -s OQ -l 4 -h OQoOcHTS.8Eiw
password found = paco

real    0m18.556s
user    0m11.580s
sys     0m6.644s
```

9.

```
localhost:~/Downloads # ./hashtime -l 4 -p 1000 -m des
des mean value: 0.002629 miliseconds
localhost:~/Downloads # ./hashtime -l 6 -p 1000 -m des
des mean value: 0.003494 miliseconds
localhost:~/Downloads # ./hashtime -l 8 -p 1000 -m des
des mean value: 0.00346 miliseconds

localhost:~/Downloads # ./hashtime -l 4 -p 1000 -m md5
md5 mean value: 0.158895 miliseconds
localhost:~/Downloads # ./hashtime -l 6 -p 1000 -m md5
md5 mean value: 0.15626 miliseconds
localhost:~/Downloads # ./hashtime -l 8 -p 1000 -m md5
md5 mean value: 0.153476 miliseconds

localhost:~/Downloads # ./hashtime -l 4 -p 1000 -m sha-256
sha-256 mean value: 3.33172 miliseconds
localhost:~/Downloads # ./hashtime -l 6 -p 1000 -m sha-256
sha-256 mean value: 3.33976 miliseconds
localhost:~/Downloads # ./hashtime -l 8 -p 1000 -m sha-256
sha-256 mean value: 3.67951 miliseconds

localhost:~/Downloads # ./hashtime -l 4 -p 1000 -m sha-512
sha-512 mean value: 2.83639 miliseconds
localhost:~/Downloads # ./hashtime -l 6 -p 1000 -m sha-512
sha-512 mean value: 2.75942 miliseconds
localhost:~/Downloads # ./hashtime -l 8 -p 1000 -m sha-512
sha-512 mean value: 2.9751 miliseconds
```

10. $T = L^{80} \cdot media$
    - DES
        - $L = 4 \rightarrow T = 4^{80} \cdot 0.002629 = 3.842D + 45$
        - $L = 6 \rightarrow T = 6^{80} \cdot 0.003494 = 6.243D + 59$
        - $L = 8 \rightarrow T = 8^{80} \cdot 0.00346 = 6.113D + 69$
    - MD5
        - $L = 4 \rightarrow T = 4^{80} \cdot 0.158895 = 2.322D + 47$
        - $L = 6 \rightarrow T = 6^{80} \cdot 0.15626 = 2.792D + 61$
        - $L = 8 \rightarrow T = 8^{80} \cdot 0.153476 = 2.712D + 71$
    - SHA-256
        - $L = 4 \rightarrow T = 4^{80} \cdot 3.33172 = 4.869D + 48$
        - $L = 6 \rightarrow T = 6^{80} \cdot 3.33976 = 5.968D + 62$
        - $L = 8 \rightarrow T = 8^{80} \cdot 3.67915 = 6.500D + 72$
    - SHA-512
        - $L = 4 \rightarrow T = 4^{80} \cdot 2.83639 = 4.145D + 48$
        - $L = 6 \rightarrow T = 6^{80} \cdot 2.75942 = 4.931D + 62$
        - $L = 8 \rightarrow T = 8^{80} \cdot 2.9751 = 5.257D + 72$
11. $1.892D + 56$ para tardar 1 año