

Práctica 10. Gestión de contraseñas

Seguridad de Sistemas Informáticos

Cheuk Kelly Ng Pante (alu0101364544@ull.edu.es)

7 de diciembre de 2023

Índice general

1. Instalación de paquetes	1
2. Generar un fichero de contraseñas con hashes MD5	2
3. Creacion de usuarios con contraseñas generadas por <i>pwgen</i>	3
4. Crackear las contraseñas con la herramienta <i>John the Ripper</i>	5
5. Bibliografía	6

1. Instalación de paquetes

- **pwgen**. Generador de contraseñas fuertes: `sudo apt-get install pwgen`

```
usuario@MV-p10:~$ sudo apt install pwgen
[sudo] contraseña para usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  pwgen
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 19,6 kB de archivos.
Se utilizarán 52,2 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 pwgen amd64 2.08-2 [19,6 kB]
Descargados 19,6 kB en 0s (122 kB/s)
Seleccionando el paquete pwgen previamente no seleccionado.
(Leyendo la base de datos ... 43557 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../pwgen_2.08-2_amd64.deb ...
Desempaquetando pwgen (2.08-2) ...
Configurando pwgen (2.08-2) ...
Procesando disparadores para man-db (2.11.2-2) ...
usuario@MV-p10:~$
```

Figura 1.1: Instalación de pwgen

- **makepasswd**. Generador de contraseñas aleatorias seguras y fiables: `sudo apt-get install makepasswd`

```
usuario@MV-p10:~$ sudo apt install makepasswd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libbytes-random-secure-perl libcrypt-passwdmd5-perl libcrypt-random-seed-perl libmath-random-isaac-perl
  libmath-random-isaac-xs-perl
Se instalarán los siguientes paquetes NUEVOS:
  libbytes-random-secure-perl libcrypt-passwdmd5-perl libcrypt-random-seed-perl libmath-random-isaac-perl
  libmath-random-isaac-xs-perl makepasswd
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 106 kB de archivos.
Se utilizarán 301 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Figura 1.2: Instalación de makepasswd

- **apg**. Generador automático de contraseñas: `sudo apt-get install apg`

```
usuario@MV-p10:~$ sudo apt install apg
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  apg
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 52,7 kB de archivos.
Se utilizarán 145 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 apg amd64 2.2.3.dfsg.1-5+b2 [52,7 kB]
Descargados 52,7 kB en 0s (363 kB/s)
Seleccionando el paquete apg previamente no seleccionado.
(Leyendo la base de datos ... 43632 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apg_2.2.3.dfsg.1-5+b2_amd64.deb ...
Desempaquetando apg (2.2.3.dfsg.1-5+b2) ...
Configurando apg (2.2.3.dfsg.1-5+b2) ...
Procesando disparadores para man-db (2.11.2-2) ...
```

Figura 1.3: Instalación de apg

- **john (John The Ripper)**. Crackeador de contraseñas: `sudo apt-get install john`

```
usuario@MV-p10:~$ sudo apt install john
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  john-data
Se instalarán los siguientes paquetes NUEVOS:
  john john-data
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 8.964 kB de archivos.
Se utilizarán 20,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Figura 1.4: Instalación de john

2. Generar un fichero de contraseñas con hashes MD5

Para generar un fichero de contraseñas con hashes MD5, se utilizará la herramienta *makepasswd* pero como se está usando una máquina virtual del IAAS de la ULL y estas máquinas tienen baja entropía en el generador de números aleatorios, hay que instalar y configurar *haveged* para resolverlo. Para ello, se ejecuta el siguiente comando: `sudo apt-get install haveged` y configuramos el fichero `/etc/default/haveged` asegurando que la variable `DAEMON_ARGS` contenga lo siguiente: `DAEMON_ARGS="-w 1024"`. Una vez hecho esto, hay que estar seguro de que el servicio *haveged* se está ejecutando: `update-rc.d haveged defaults`

Ahora, se puede generar el fichero de contraseñas con hashes MD5 con el siguiente comando:

```
1 echo "mypassword" | makepasswd --clearfrom=- --crypt-md5 | awk '{ print $2 }'
```

```
usuario@MV-p10:~$ echo "mypassword" | makepasswd --clearfrom=- --crypt-md5 | awk '{ print $2 }'  
$1$CWB5g.e8$ErLD5H/MR0dThDZ7ebWAZ.  
usuario@MV-p10:~$
```

Figura 2.1: Generación de contraseñas con hashes MD5

3. Creacion de usuarios con contraseñas generadas por *pwgen*

Para crear usuarios con contraseñas generadas por *pwgen*, se ha creado un script que genera cinco usuarios con contraseñas de menor a mayor fortaleza. El script es el siguiente:

```
1  #!/bin/bash
2
3  USERS=("usuario1" "usuario2" "usuario3" "usuario4" "usuario5")
4
5  PASSWORDS=()
6
7  for user in "${USERS[@]"; do
8      sudo useradd $user
9      if [ $user == "usuario1" ]; then
10         password1="usuario1"
11         password1_md5=$(echo $password1 | makepasswd --clearfrom=- --crypt-md5 | awk '{
12             print $2 }')
13         echo "$user:$password1" | sudo chpasswd
14         PASSWORDS+=("$password1")
15     elif [ $user == "usuario2" ]; then
16         password2="123456789"
17         password2_md5=$(echo $password2 | makepasswd --clearfrom=- --crypt-md5 | awk '{
18             print $2 }')
19         echo "$user:$password2" | sudo chpasswd
20         PASSWORDS+=("$password2")
21     elif [ $user == "usuario3" ]; then
22         password3="UsuArio3"
23         password3_md5=$(echo $password3 | makepasswd --clearfrom=- --crypt-md5 | awk '{
24             print $2 }')
25         echo "$user:$password3" | sudo chpasswd
26         PASSWORDS+=("$password3")
27     elif [ $user == "usuario4" ]; then
28         password4="_User@4_"
29         password4_md5=$(echo $password4 | makepasswd --clearfrom=- --crypt-md5 | awk '{
30             print $2 }')
31         echo "$user:$password4" | sudo chpasswd
32         PASSWORDS+=("$password4")
33     elif [ $user == "usuario5" ]; then
34         password5=$(pwgen -y -A 8 -1)
35         password5_md5=$(echo $password5 | makepasswd --clearfrom=- --crypt-md5 | awk '{
36             print $2 }')
37         echo "$user:$password5" | sudo chpasswd
38         PASSWORDS+=("$password5")
39     fi
40 done
```

Al ejecutar el script, se crean los usuarios con las contraseñas generadas por *pwgen*:

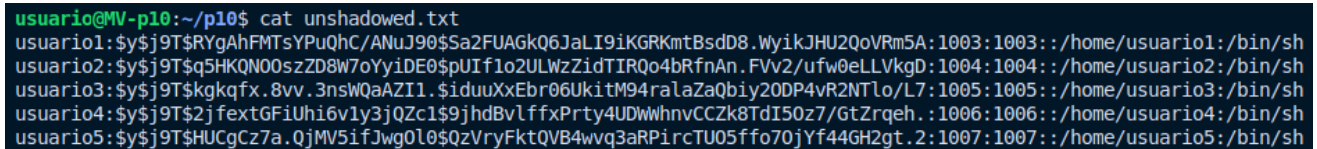
```
Usuarios y contraseñas creados:
-----
Usuario: usuario1 | Contraseña: phaikahc
Usuario: usuario2 | Contraseña: Roa6fohj9t
Usuario: usuario3 | Contraseña: ha7it[i3
Usuario: usuario4 | Contraseña: oG=o'ephu4
Usuario: usuario5 | Contraseña: EmT9KtRMstAyNiML
```

Figura 3.1: Creación de usuarios con contraseñas generadas por *pwgen*

Una vez creado los usuarios, vamos a proceder a hacer un paso básico que es el "desombreado" que es un proceso en el que se cambian el fichero `/etc/passwd` y `/etc/shadow` para que el hash de la contraseña se encuentre en el fichero `/etc/passwd` y no en el fichero `/etc/shadow`. Para ello, se ejecuta el siguiente comando:

```
1 sudo unshadow passwd.txt shadow.txt > unshadowed.txt
```

Ahora el fichero `unshadowed.txt` tendrá el siguiente contenido:

A terminal window with a dark background and light green text. The prompt is 'usuario@MV-p10:~/p10\$'. The command 'cat unshadowed.txt' has been executed, displaying five lines of user data. Each line contains a username, a long alphanumeric hash, a UID:GID format, and a home directory with shell path.

```
usuario@MV-p10:~/p10$ cat unshadowed.txt
usuario1:$y$j9T$RYgAhFMTsYPuQhC/ANuJ90$Sa2FUAGkQ6JaLI9iKGRKmtBsdD8.WyikJHU2QoVRm5A:1003:1003:/home/usuario1:/bin/sh
usuario2:$y$j9T$q5HKQN00szZD8W7oYyiDE0$pUIflo2ULWzZidTIRQo4bRfnAn.FVv2/ufw0eLLVkgD:1004:1004:/home/usuario2:/bin/sh
usuario3:$y$j9T$kgkqfx.8vv.3nsWQaAZI1.$iduuXxEbr06UkitM94ralaZaQbiy20DP4vR2NTlo/L7:1005:1005:/home/usuario3:/bin/sh
usuario4:$y$j9T$2jfxTGFiUhi6v1y3jQZcl$9jhdBvlfxfPrty4UDWWhnvCCZk8TdI50z7/GtZrqeh.:1006:1006:/home/usuario4:/bin/sh
usuario5:$y$j9T$HUCgCz7a.QjMV5ifJwg0l0$QzVryFktQVB4wvq3aRPircTU05ffo70jYf44GH2gt.2:1007:1007:/home/usuario5:/bin/sh
```

Figura 3.2: Contenido del fichero `unshadowed.txt`

4. Crackear las contraseñas con la herramienta *John the Ripper*

5. Bibliografía

1. LaMendola, S. (2013). How to Setup Additional Entropy for Cloud Servers Using Haveged. DigitalOcean. Recuperado de <https://www.digitalocean.com/community/tutorials/how-to-setup-additional-entropy-for-cloud-servers-using-haveged>
2. erev0s. (2020). Cracking /etc/shadow with John. Recuperado de <https://erev0s.com/blog/cracking-etcshadow-john/>