

Práctica 10. Gestión de contraseñas

Seguridad de Sistemas Informáticos

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

11 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 | 7 |

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 }'
```

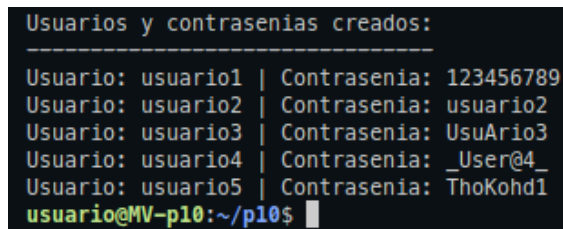
```
usuario@MV-p10:~$ echo "mypassword" | makepasswd --clearfrom=- --crypt-md5 | awk '{ print $2 }'  
$1$CWB5g.e8$ErLD5H/MR0dThDZ7ebWAZ.  
usuario@MV-p10:~$
```

Figura 2.1: Generación de contraseñas con hashes MD5

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

Para crear los usuarios se ha optado por crear un script que cree los usuarios y que asigne las contraseñas con hashes MD5. 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 [ $user == "usuario1" ]; then
10         password1="123456789"
11         password1_md5=$(echo $password1 | makepasswd --clearfrom=- --crypt-md5 | awk '{
12             print $2 }')
13         echo "$user:$password1" | sudo chpasswd
14         PASSWORDS+=("$password1")
15     elif [ $user == "usuario2" ]; then
16         password2="usuario2"
17         password2_md5=$(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
```



```
Usuarios y contraseñas creados:
-----
Usuario: usuario1 | Contraseña: 123456789
Usuario: usuario2 | Contraseña: usuario2
Usuario: usuario3 | Contraseña: UsuArio3
Usuario: usuario4 | Contraseña: _User@4_
Usuario: usuario5 | Contraseña: ThoKohd1
usuario@MV-p10:~/p10$
```

Figura 3.1: Resultado de ejecución del script

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 combinan 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, haremos copias de estos archivos y los llamaremos `passwd` y `shadow` respectivamente y lo haremos en el directorio donde se vaya a trabajar. Una vez hecho esto, se ejecuta el siguiente comando:

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

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

```
usuario@MV-p10:~/p10/prueba_john$ cat unshadowed.txt
usuario1:$y$j9T$AiVaCfdep0ZG.BuM6xFlk.$8V6l4XIBUQwi2L8fu0.zixNZGPK79AnqBKuo5CGho.2:1003:1003:./home/usuario1:/bin/sh
usuario2:$y$j9T$7M4hAqtOzN/8w7/v3yvJd.$s/0FfR8UDnjhN/n01Huy2ZD4hw1zrsAiX/0W3XWIZk4:1004:1004:./home/usuario2:/bin/sh
usuario3:$y$j9T$TH30tG6eTYc0mdn0p.C071$vgwxA4QZfwk01Zymx0CBEozThEn0npjXYTd1Nx3wkb2:1005:1005:./home/usuario3:/bin/sh
usuario4:$y$j9T$U60f8lz22DCOWkTFERIBm.$I21CqHN670/XhS4IfJlF3CT07RyXRLkK.VXfSzs0Zr8:1006:1006:./home/usuario4:/bin/sh
usuario5:$y$j9T$RtJ9nUIEKV73eV59By7p1$PXA VHfXDebMJd/t8sxdreQmw0d38Ni4q7wjwZ3TG5x5:1007:1007:./home/usuario5:/bin/sh
usuario@MV-p10:~/p10/prueba_john$
```

Figura 3.2: Contenido del fichero `unshadowed.txt`

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

Para crackear las contraseñas con la herramienta *John the Ripper*, según los modos, se ejecutan los siguientes comandos:

- **Invocación básica:** `john --format=crypt unshadowed.txt`

```
usuario@MV-p10:~/p10$ sudo john --format=crypt unshadowed.txt
[sudo] contraseña para usuario:
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Will run 24 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
usuario2          (usuario2)
1g 0:00:00:05 72% 1/3 0.1686g/s 355.4p/s 356.1c/s 356.1C/s Usuario323..usuario363
123456789        (usuario1)
2g 0:00:00:45 4% 2/3 0.04362g/s 188.1p/s 418.4c/s 418.4C/s Devine..Hanna
2g 0:00:01:23 7% 2/3 0.02382g/s 172.4p/s 437.7c/s 437.7C/s kimberly1..reynolds1
2g 0:00:01:25 7% 2/3 0.02326g/s 172.9p/s 438.7c/s 438.7C/s dexter1..hedgehog1
2g 0:00:01:27 7% 2/3 0.02273g/s 172.2p/s 439.5c/s 439.5C/s perfect1..skyer1
2g 0:00:01:29 8% 2/3 0.02225g/s 171.8p/s 439.9c/s 439.9C/s steph1..bonital
2g 0:05:04:37 3/3 0.000109g/s 155.8p/s 467.0c/s 467.0C/s asawtz..asshk5
2g 0:12:36:51 3/3 0.000044g/s 155.6p/s 466.8c/s 466.8C/s sevly20..sevlo25
2g 0:12:36:54 3/3 0.000044g/s 155.6p/s 466.8c/s 466.8C/s socutik..socusic
2g 0:13:14:19 3/3 0.000041g/s 155.6p/s 466.8c/s 466.8C/s subz29..subzys
2g 0:20:18:09 3/3 0.000027g/s 155.6p/s 466.9c/s 466.9C/s 27mak5..27mk25
2g 0:21:23:57 3/3 0.000025g/s 155.6p/s 466.9c/s 466.9C/s jajnlc..jostyb
2g 1:12:47:13 3/3 0.000015g/s 155.4p/s 466.4c/s 466.4C/s mrufo..msl.d
2g 2:00:55:32 3/3 0.000011g/s 154.4p/s 463.2c/s 463.2C/s culesami..culeid14
2g 2:00:57:24 3/3 0.000011g/s 154.4p/s 463.2c/s 463.2C/s 12563435..12444855
```

Figura 4.1: Invocación básica de *John the Ripper*

- **Invocación con diccionario:** `john --format=crypt -w:crackstation-human-only.txt unshadowed.txt`

Para este caso, se ha usado el diccionario *crackstation-human-only.txt* que se puede encontrar en <https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm> y es una lista que contiene todas las listas de palabras, diccionarios y bases de datos de contraseñas que la persona que lo hizo ha recopilado de Internet. También contiene todas las palabras de las bases de datos de Wikipedia (páginas-artículos, recuperados en 2010, todos los idiomas), así como montones de libros del Proyecto Gutenberg. También incluye las contraseñas de algunas violaciones de bases de datos de perfil bajo que se vendían en la deep-weeb hace años.

```
usuario@MV-p10:~/p10/prueba_john$ sudo john --format=crypt -w:crackstation-human-only.txt unshadowed.txt
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Will run 24 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:04 0% 0g/s 46.37p/s 231.8c/s 231.8C/s !!!ANGELBABY...!!!fezil3327!!!
0g 0:00:00:06 0% 0g/s 46.75p/s 233.7c/s 233.7C/s !!!fish!!!...!!!loveuu
```

Figura 4.2: Invocación con diccionario de *John the Ripper*

- **Invocación con wordlist:** `john --format=crypt --wordlist=crackstation-human-only.txt unshadowed.txt` En esta invocación, se ha usado el mismo diccionario que en la invocación anterior.

```

usuario@MV-p10:~/p10/john_wordlist$ sudo john --format=crypt --wordlist=crackstation-human-only.txt unshadowed.txt
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Remaining 4 password hashes with 4 different salts
Will run 24 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:29 0% 0g/s 57.79p/s 240.8c/s 240.8C/s !!LArue100..!!lily!!

```

Figura 4.3: Invocación con wordlist de *John the Ripper*

- **Invocación single:** `john --format=crypt --single unshadowed.txt`

```

usuario@MV-p10:~/p10/prueba_john$ sudo john --format=crypt --single unshadowed.txt
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Remaining 4 password hashes with 4 different salts
Will run 24 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
usuario2 (usuario2)
1g 0:00:00:07 81% 0.1267g/s 230.7p/s 231.1c/s 231.1C/s usuario564..Usuario5555
1g 0:00:00:12 100% 0.07911g/s 219.3p/s 234.4c/s 234.4C/s usuario41996..usuario41900
Use the "--show" option to display all of the cracked passwords reliably
Session completed
usuario@MV-p10:~/p10/prueba_john$ sudo john --format=crypt --single unshadowed.txt
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Remaining 3 password hashes with 3 different salts
Will run 24 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:08 81% 0g/s 239.7p/s 239.7c/s 239.7C/s usuario465..Usuario4666
0g 0:00:00:12 100% 0g/s 220.6p/s 235.9c/s 235.9C/s usuario41997..usuario41900
Session completed

```

Figura 4.4: Invocación single de *John the Ripper*

- **Invocación incremental:** `john --format=crypt --incremental unshadowed.txt`

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/>
3. DefuseSec. (2019). CrackStation's Password Cracking Dictionary. Recuperado de <https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm>