John the Ripper

Práctica de Contraseñas

Tabla de contenido

Tabla de contenido..... 1
Unidad didáctica: Práctica de contraseñas..... 2
Linux 2
1.- Comprueba la fortaleza de las claves de tu sistema..... 2
Windows..... 9
2.- Comprueba la fortaleza de las claves de tu sistema: 9

Unidad didáctica: Práctica de contraseñas.

Linux

1.- Comprueba la fortaleza de las claves de tu sistema

John the Ripper, es un programa que nos permite recuperar contraseñas a partir de los usuarios que existen en nuestro sistema.

Aunque realmente es una herramienta para comprobar la complejidad de nuestras contraseñas.

Para comprobar la calidad de nuestras contraseñas en Ubuntu, primero actualizaremos nuestros repositorios. Con el comando:

apt-get update

```
root@ubuntucliente:/home/usuario# apt-get update
Ign http://security.ubuntu.com natty-security InRelease
Des:1 http://security.ubuntu.com natty-security Release.gpg [198 B]
Ign http://extras.ubuntu.com natty InRelease
Ign http://es.archive.ubuntu.com natty InRelease
Des:2 http://security.ubuntu.com natty-security Release [31,4 kB]
Des:3 http://extras.ubuntu.com natty Release.gpg [72 B]
Ign http://es.archive.ubuntu.com natty-updates InRelease
Des:4 http://extras.ubuntu.com natty Release [9753 B]
Des:5 http://security.ubuntu.com natty-security/main Sources [72,3 kB]
Des:6 http://extras.ubuntu.com natty/main Sources [14 B]
Des:7 http://extras.ubuntu.com natty/main amd64 Packages [14 B]
Ign http://extras.ubuntu.com natty/main TranslationIndex
99% [5 Sources bzip2 0 B] [Esperando las cabeceras] [Esperando las cabeceras] [
```

Instalamos John the Ripper, ejecutando el comando:

apt-get install john

```
root@ubuntucliente:/home/usuario# apt-get install john
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  john-data
Se instalarán los siguientes paquetes NUEVOS:
  john john-data
0 actualizados, 2 se instalarán, 0 para eliminar y 247 no actualizados.
Necesito descargar 862 kB de archivos.
Se utilizarán 1622 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Las últimas versiones de Ubuntu para encriptar las claves utilizan sha-512, por ello debemos compilar John the Ripper, para poder utilizarlo en cualquier otra distribución.

Primer paso será descargar el paquete desde la página con el comando wget, como muestra la imagen:

```
root@ubuntucliente:/home/usuario# wget http://www.openwall.com/john/g/john-1.7.8
-jumbo-7.tar.gz
--2011-10-19 13:58:47--  http://www.openwall.com/john/g/john-1.7.8-jumbo-7.tar.g
z
Resolviendo www.openwall.com... 195.42.179.202
Conectando a www.openwall.com|195.42.179.202|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1342588 (1,3M) [application/x-tar]
Guardando en: «john-1.7.8-jumbo-7.tar.gz»

100%[======================================>] 1.342.588    332K/s   en 5,5s

2011-10-19 13:58:54 (236 KB/s) - «john-1.7.8-jumbo-7.tar.gz» guardado [1342588/1
342588]

root@ubuntucliente:/home/usuario#
```

Descomprimimos el fichero con el comando tar y las opciones xvzf

```
root@ubuntucliente:/home/usuario# tar xvzf john-1.7.8-jumbo-7.tar.gz
john-1.7.8-jumbo-7/
john-1.7.8-jumbo-7/doc/
john-1.7.8-jumbo-7/doc/CHANGES
john-1.7.8-jumbo-7/doc/CONFIG
john-1.7.8-jumbo-7/doc/CONTACT
john-1.7.8-jumbo-7/doc/CREDITS
john-1.7.8-jumbo-7/doc/ENCODINGS
john-1.7.8-jumbo-7/doc/EPi.patch.README
john-1.7.8-jumbo-7/doc/EXAMPLES
john-1.7.8-jumbo-7/doc/EXTERNAL
john-1.7.8-jumbo-7/doc/FAQ
john-1.7.8-jumbo-7/doc/HDAA_README
```

Entramos al directorio del fichero descomprimido y específicamente al directorio run y dentro listamos el contenido

```
root@ubuntucliente:/home/usuario# cd john-1.7.8-jumbo-7
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7# cd run/
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# ls
all.chr     digits.chr    genincstats.rb  mailer        sap_prepare.pl
alnum.chr   dumb16.conf   john.conf       netntlm.pl    sha-dump.pl
alpha.chr   dumb32.conf   lanman.chr      netscreen.py  sha-test.pl
cmpt_cp.pl  generic.conf  ldif2pw.pl      password.lst  stats
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run#
```

Dentro del contenido podemos observar, que no tenemos los ficheros ejecutables enlaces, etc. Necesarios para comprobar las contraseñas.

Por eso, retrocederemos en la ruta, e ingresaremos al directorio src, y ejecutamos el comando make, al igual en la imagen.

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7# cd src/
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src# make
To build John the Ripper, type:
        make clean SYSTEM
where SYSTEM can be one of the following:
linux-x86-64            Linux, x86-64 with SSE2 (best tested)
linux-x86-64-avx        Linux, x86-64 with AVX (experimental)
linux-x86-64-xop        Linux, x86-64 with AVX and XOP (experimental)
linux-x86-64-icc        Linux, x86-64 compiled with icc (best)
linux-x86-64-clang      Linux, x86-64 compiled with clang (good)
linux-x86-sse2          Linux, x86 32-bit with SSE2 (best tested if 32-bit)
linux-x86-sse2i         Linux, x86 32-bit with SSE2 (32-bit, intrinsic)
linux-x86-mmx           Linux, x86 32-bit with MMX (for old computers)
linux-x86-any           Linux, x86 32-bit (for truly ancient computers)
linux-x86-avx           Linux, x86 32-bit with AVX (experimental)
linux-x86-xop           Linux, x86 32-bit with AVX and XOP (experimental)
```

Buscamos el paquete que necesitamos para compilar, en nuestro caso es una versión de 64bits, elegimos el Linux-x86-64 with SSE2

Una vez localizado el paquete, ejecutamos:

make clean Linux-x86-64

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src# make clean linux-x86-64
rm -f ../run/john ../run/unshadow ../run/unafs ../run/unique ../run/undrop ../ru
n/ssh2john ../run/pdf2john ../run/rar2john ../run/zip2john ../run/genmkvpwd ../r
un/mkvcalcproba ../run/calc_stat ../run/tgtsnarf ../run/john.bin ../run/john.com
 ../run/unshadow.com ../run/unafs.com ../run/unique.com ../run/undrop.com ../run
/ssh2john.com ../run/pdf2john.com ../run/rar2john.com ../run/zip2john ../run/joh
n.exe ../run/unshadow.exe ../run/unafs.exe ../run/unique.exe ../run/undrop.exe .
./run/ssh2john.exe ../run/pdf2john.exe ../run/rar2john.exe ../run/zip2john.exe .
./run/genmkvpwd.exe ../run/mkvcalcproba.exe ../run/calc_stat.exe ../run/john-min
gw.exe ../run/unshadow.exe ../run/unafs.exe ../run/unique.exe ../run/undrop.exe
../run/ssh2john.exe ../run/pdf2john.exe ../run/rar2john.exe ../run/zip2john.exe
../run/genmkvpwd.exe ../run/mkvcalcproba.exe ../run/calc_stat.exe
rm -f ../run/john.exe john-macosx-* *.o *.bak core
```

Si al terminar la compilación aparece un error, es por la falta de una librería.

```
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops md5_eq.c
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops md5.c
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops rc4.c
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops hmacmd5.c
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops base64.c
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops md4.c
gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include  -DHAVE_CRYPT -DHAVE_
DL -funroll-loops md5_gen_fmt.c
In file included from md5_gen_fmt.c:145:0:
sha.h:4:25: fatal error: openssl/sha.h: No existe el fichero o el directorio
compilation terminated.
make[1]: *** [md5_gen_fmt.o] Error 1
make[1]: se sale del directorio «/home/usuario/john-1.7.8-jumbo-7/src»
make: *** [linux-x86-64] Error 2
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src#
```

Dentro de nuestro repositorio buscamos el paquete necesario, el cual es libssl-dev

Lo haremos ejecutando el siguiente comando:

apt-cache search openssl | grep dev

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src# apt-cache search openss
l | grep dev
libghc6-sha-dev - Haskell SHA suite of message digest functions - GHC 6 librarie
s
libglobus-gsi-openssl-error-dev - Globus Toolkit - Globus OpenSSL Error Handling
 Development Files
libglobus-openssl-dev - Globus Toolkit - Openssl Library Development Files
libglobus-openssl-module-dev - Globus Toolkit - Globus OpenSSL Module Wrapper De
velopment Files
liblwt-ssl-ocaml-dev - cooperative OpenSSL bindings for OCaml
libpion-common-dev - lightweight HTTP interface library - common development fil
es
libpion-net-dev - lightweight HTTP interface library - development files
libssl-ocaml-dev - OCaml bindings for OpenSSL
libssl-dev - bibliotecas de desarrollo SSL, cabecera y documentación
libpathfinder-dev - Archivos de desarrollo para pathfinder
libcurl4-openssl-dev - Development files and documentation for libcurl (OpenSSL)
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src#
```

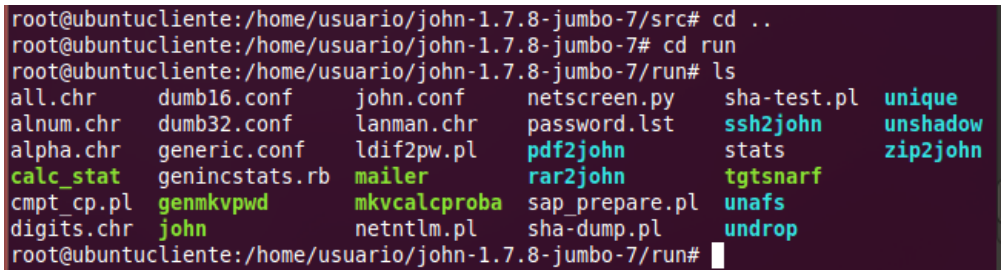
Una vez, que conocemos el paquete, pasamos a instalarlo:

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src# apt-get install libssl-
dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  zlib1g-dev
Se instalarán los siguientes paquetes NUEVOS:
  libssl-dev zlib1g-dev
0 actualizados, 2 se instalarán, 0 para eliminar y 247 no actualizados.
Necesito descargar 2343 kB de archivos.
Se utilizarán 7713 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

Termina la instalación de la librería, ya podemos volver a compilar, como podemos ver en la imagen:

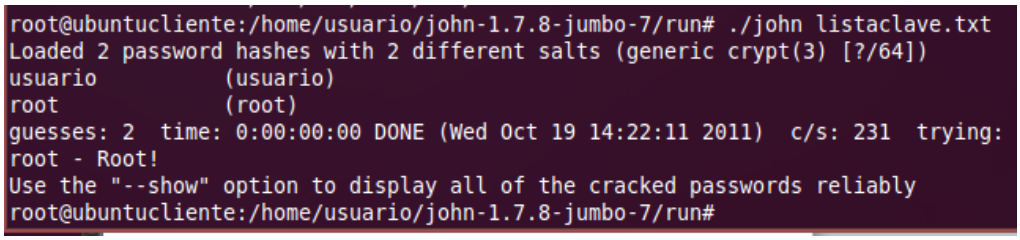
```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/src# make clean linux-x86-64
rm -f ../run/john ../run/unshadow ../run/unafs ../run/unique ../run/undrop ../ru
n/ssh2john ../run/pdf2john ../run/rar2john ../run/zip2john ../run/genmkvpwd ../r
un/mkvcalcproba ../run/calc_stat ../run/tgtsnarf ../run/john.bin ../run/john.com
 ../run/unshadow.com ../run/unafs.com ../run/unique.com ../run/undrop.com ../run
/ssh2john.com ../run/pdf2john.com ../run/rar2john.com ../run/zip2john ../run/joh
n.exe ../run/unshadow.exe ../run/unafs.exe ../run/unique.exe ../run/undrop.exe .
./run/ssh2john.exe ../run/pdf2john.exe ../run/rar2john.exe ../run/zip2john.exe .
./run/genmkvpwd.exe ../run/mkvcalcproba.exe ../run/calc_stat.exe ../run/john-min
gw.exe ../run/unshadow.exe ../run/unafs.exe ../run/unique.exe ../run/undrop.exe
../run/ssh2john.exe ../run/pdf2john.exe ../run/rar2john.exe ../run/zip2john.exe
../run/genmkvpwd.exe ../run/mkvcalcproba.exe ../run/calc_stat.exe
rm -f ../run/john.exe john-macosx-* *.o *.bak core
```

Retrocedemos de la carpeta y entramos al directorio run y lo listamos, y vemos ya los ficheros necesarios para comprobar la complejidad de nuestras contraseñas.



A continuación, combinamos los ficheros /etc/passwd/ y /etc/shadow con el comando unshadow en un fichero listaclave.txt, y luego visualizamos el contenido del fichero utilizando el comando tail, al igual que la imagen:

Paso seguido ejecutaremos John sobre el fichero listaclave.txt, como vemos en la imagen:



John the Ripper va probando el modo single crack, pasando a usar un diccionario con reglas y por último, el modo incremental, de forma que si la contraseña de un usuario es débil la encontrará en segundos.

Si deseamos probar con usuario nuevos, primero creamos un nuevo usuario con el comando adduser y le asignamos una contraseña débil 1234.

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# adduser a
Añadiendo el usuario `a' ...
Añadiendo el nuevo grupo `a' (1001) ...
Añadiendo el nuevo usuario `a' (1001) con grupo `a' ...
Creando el directorio personal `/home/a' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para a
Introduzca el nuevo valor, o presione ENTER para el predeterminado
        Nombre completo []:
        Número de habitación []:
        Teléfono del trabajo []:
        Teléfono de casa []:
        Otro []:
¿Es correcta la información? [S/n] s
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run#
```

Volvemos a combinar los fichero /etc/passwd /etc/shadow sobre el fichero lista-clave.txt con el comando unshadow.

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# ./unshadow /etc/passwd
/etc/shadow > listaclave.txt
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# tail listaclave.txt
gdm:*:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
speech-dispatcher:!:107:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
sh
kernoops:*:108:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:*:109:116:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:*:110:119:RealtimeKit,,,:/proc:/bin/false
hplip:*:111:7:HPLIP system user,,,:/var/run/hplip:/bin/false
saned:*:112:121::/home/saned:/bin/false
usuario:$6$VgtX0FbB$rHqvOj4LETQRnUeJcNmEWD36sJgNQtDdki29Ozs95SqRbKu4hK3BWNpqjmOi
cZzUNqLQMz/.LvIzN8y/HK9QZ1:1000:1000:usuario,,,:/home/usuario:/bin/bash
statd:*:113:65534::/var/lib/nfs:/bin/false
a:$6$st0/N0jb$iTyB0BGH6iCQv8spthK.y9CTQXVDkNx0SWMguQGpV387ehMU7.SzWxGBij2ZnnRR33
1sEWoWSKybqh8TE2iqE/:1001:1001:,,,:/home/a:/bin/bash
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run#
```

Y por últimos John.

```
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# ./john listaclave.txt
Loaded 3 password hashes with 3 different salts (generic crypt(3) [?/64])
Remaining 1 password hash
1234             (a)
guesses: 1  time: 0:00:00:03 DONE (Wed Oct 19 14:35:51 2011)  c/s: 232  trying:
12345 - missy
Use the "--show" option to display all of the cracked passwords reliably
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run#
```

Las contraseñas descifradas se almacenan en el fichero John.pot, por si en un futuro queremos ver el resultado de la desencriptacion.


root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# ls
all.chr generic.conf ldif2pw.pl rar2john undrop
alnum.chr genincstats.rb listaclave.txt sap_prepare.pl unique
alpha.chr genmkvpwd mailer sha-dump.pl unshadow
calc_stat john mkvcalcproba sha-test.pl zip2john
cmpt_cp.pl john.conf netntlm.pl ssh2john
digits.chr john.log netscreen.py stats
dumb16.conf john.pot password.lst tgtsnarf
dumb32.conf lanman.chr pdf2john unafs
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run# cat john.pot
6VgtX0FbB$rHqvOj4LETQRnUeJcNmEWD36sJgNQtDdki29Ozs95SqRbKu4hK3BWNpqjmOicZzUNqLQ
Mz/.LvIzN8y/HK9QZ1:usuario
6wab3B6Rm$aeZUOU7PjNH3TBxK5vIdIKrZn9mYBk5/WEVZMfhH8DhbTu38m6hBxGjpqP9PkTTxSs9D
z8YlqpIMR4lW0FbpB/:root
6st0/N0jb$iTyB0BGH6iCQv8spthK.y9CTQXVDkNx0SWMguQGpV387ehMU7.SzWxGBij2ZnnRR331s
EWoWSKybqh8TE2iqE/:1234
root@ubuntucliente:/home/usuario/john-1.7.8-jumbo-7/run#

Windows

2.- Comprueba la fortaleza de las claves de tu sistema:

Una vez comprobado la complejidad de las contraseñas en Ubuntu, lo haremos en Windows XP.

El primer paso será descargar los dos ficheros necesarios, desde las siguientes urls:

http://www.openwall.com/john/contrib/john-1.7.6-jumbo-9-win32.zip

http://www.tarasco.org/security/pwdump_7/pwdump7.zip

Finalizada la descarga, descomprimimos los ficheros en el escritorio.

Abrimos la consola de símbolo de sistema, y entramos al directorio pwdump7, dentro del ejecutamos lo siguiente:

Pwdump7.exe > contraseñas.txt

```
C:\Documents and Settings\Andre\Escritorio>cd PwDump7

C:\Documents and Settings\Andre\Escritorio\PwDump7>PwDump7.exe > contraseña.tt
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es
```

Una vez hecho eso, copiamos el fichero contraseñas.txt en el directorio run, que está dentro de directorio John.

En el símbolo de sistema, nos vamos al directorio run, ejecutamos:

John contraseñas.txt

```
C:\Documents and Settings\Andre\Escritorio\PwDump7>cd ..

C:\Documents and Settings\Andre\Escritorio>cd john

C:\Documents and Settings\Andre\Escritorio\john>cd run

C:\Documents and Settings\Andre\Escritorio\john\run>john contraseña.txt
Loaded 5 password hashes with no different salts (LM DES [128/128 BS SSE2])
12345            (usuario)
```

Y nos desencriptará las contraseñas del fichero.

Al igual que Ubuntu, las contraseñas descifradas se almacenan en el fichero John.pot