Actividad 03. Uso de John the Ripper

John The Ripper es una herramienta de código abierto que viene instalada por defecto en el sistema operativo Kali Linux y que **sirve para descifrar contraseñas de usuarios a partir de sus códigos hash**.

Las funciones hash son irreversibles por definición y, por eso, John The Ripper funciona creando generando los códigos hash de miles de palabras incluidas en un archivo de texto plano conocido como **diccionario**. A este tipo de operación se le conoce como ataque de diccionario y John The Ripper es el programa más comúnmente utilizado para ejecutar uno.

Para aprender qué es John The Ripper, es necesario conocer sus cuatro modos de funcionamiento, que son:

- **Single crack**
- **Diccionario**
- **Incremental**
- **Externo**

Si no se especifica el modo que se quiere utilizar, el programa John The Ripper ejecutará cada uno, en este orden, por defecto.

Single crack

En este modo, **John The Ripper utiliza el nombre del usuario para generar diferentes combinaciones** y generar su código hash para compararlo con el que se está intentando romper o crackear. En muchas ocasiones, las personas utilizan su mismo nombre de usuario o una variación del mismo como contraseña. Tanto así, que esto es lo primero que prueba este *software*.

Diccionario o wordlist

El modo diccionario de **John The Ripper** implica el uso de una **lista de palabras predeterminadas** que puede adquirirse de diferentes formas. Existen diccionarios famosos como, por ejemplo, el RockYou.txt, pero, asimismo, se pueden crear listas de palabras propias a partir de bases de datos filtradas, lo cual es común en los ciberataques.

Incremental

En este modo, John The Ripper **intenta crackear la contraseña de un usuario mediante un ataque por fuerza bruta**, de acuerdo con ciertas reglas que ya han sido predefinidas por el programa.

Externo

Se utiliza un *software* externo a John, el cual debe estar escrito en lenguaje de programación C, y se utiliza para ejecutar ataques de fuerza bruta, siguiendo los lineamientos que estén inscritos en dicho programa. Es un modo con el cual se pueden hacer este tipo de ataques con lineamientos escogidos directamente por el programador.

¿Cómo funciona John The Ripper?

Veamos **cuál es su funcionamiento en el modo diccionario**, que suele ser el más utilizado.

1. Primero, puedes indicarle a John cuál es el **tipo de hash** que estás intentando romper. Si no, el programa puede identificarlo por sí solo. No obstante, especificar este dato agilizará el funcionamiento de la aplicación. Por ejemplo (para un hash en formato MD5): *–format=rawmd5*
2. Luego, debes escoger **la ruta del diccionario**. Por ejemplo: *–wordlist=/usr/share/wordlists/rockyou.txt*
3. Finalmente, debes indicar **el nombre del archivo de texto** que contiene el código hash a romper. Por ejemplo: *hashEjemplo.txt*

A partir de este ejemplo, la línea de código de la consola se vería del siguiente modo para que aprendas cómo usar John The Ripper:

john --format=rawmd5 --wordlist=/usr/share/wordlists/rockyou.txt hashEjemplo.txt

Esto hará que el programa genere las funciones hash MD5 de cada una de las palabras del diccionario y las compare con el código hash del archivo de texto «hashEjemplo.txt» **hasta encontrar una coincidencia**.

¿Cómo protegerse de un ataque de diccionario?

Ahora que sabes qué es John The Ripper y cómo se puede usar para romper códigos hash de contraseñas, seguramente te preguntarás **cómo mantenerte a salvo de un ataque de diccionario** o de uno de fuerza bruta.

Para ello, existen **tres recomendaciones** indispensables para programadores y usuarios, para que sigas aprendiendo cómo usar John The Ripper:

- Los desarrolladores deben **escoger funciones hash seguras para contraseñas**, como, por ejemplo, las SHA512 o SHA256. Algoritmos como el MD5 y el SHA-1 se pueden romper fácilmente por medio de programas de código abierto y páginas web gratuitas.
- Los usuarios no deberían escoger contraseñas demasiado fáciles y tampoco deberían reutilizarlas. Combinaciones como «**1234**«, «**0000**«, «**hola**«, «**contraseña**«, etc., son imposibles de proteger de un ataque.
- Para evitar que se rompan los hashes de contraseñas de dificultad intermedia (que son las más comunes) y proteger a los usuarios, se puede usar la técnica del *salting*, que implica **agregarle datos aleatorios a la contraseña del usuario antes de generar el hash**. Esto también impide que se hagan estos ataques de forma satisfactoria.

Casos de Uso para John the Ripper

Ahora que entiendes los diferentes modos de John, miremos a unos pocos casos de uso.

Usaremos John para descifrar tres tipos de hashes: una contraseña de Windows NTLM, una contraseña alternativa de Linux, y la contraseña para un archivo zip.

Cómo Descifrar una Contraseña de Windows

Comencemos con Windows. En Windows, los hashes de contraseña están almacenados en la **base de datos SAM**.

SAM usa el formato de hash **LM/NTLM** para las contraseñas, así que estaremos usando John para descifrar una.

Asumamos que has adquirido un hash de contraseña para un usuario de Windows.

Este es el comando para descifrarlo:

```
john --format=lm crack.txt
```

El crack.txt contendrá el hash de la contraseña. Si John no es capaz de descifrar la contraseña usando su lista de palabras por defecto, puedes usar la lista de palabras RockYou usando la bandera --wordlist.

Cómo descifrar una Contraseña de Linux

En Linux, hay dos archivos importantes guardados en la carpeta /etc: passwd y shadow.

- /etc/passwd -> almacena información como nombre de usuario, id de usuario, shell de inicio de sesión, y así sucesivamente.
- /etc/shadow -> contiene hash de contraseñas, expiración de contraseñas, y así sucesivamente.

Además del comando "john", John viene con un par de otras utilidades. Uno de ellos se llama "unshadow".

El comando unshadow combina los archivo passwd y shadow juntos en un solo archivo. Esto después puede ser usado por John para descifrar contraseñas.

Así es como usamos el comando unshadow:

```
unshadow /etc/passwd /etc/shadow > output.db
```

Este comando combinará los archivos juntos y creará un archivo output.db. Ahora podemos descifrar el archivo output.db usando John.

```
john output.db
```

John intenta encontrar la contraseña para todos los usuarios en el archivo passwd y genera la salida con la lista de contraseñas descifradas. De nuevo, puedes usar listas de palabras personalizadas con la bandera --wordlist.

Cómo descifrar una Contraseña de un archivo Zip

Finalmente, vamos a descifrar una contraseña de un archivo zip. Para hacer eso, primero tenemos que obtener el hash de la contraseña del archivo zip.

Como unshadow, John tiene otra utilidad llamado zip2john. zip2john nos ayuda en obtener el hash de los archivos zip. Si estás descifrando un archivo .rar, puedes usar la utilidad rar2john.

Esta es la sintaxis para obtener el hash de la contraseña de un archivo zip:

```
zip2john file.zip > zip.hashes
```

El comando de arriba obtendrá el hash de un archivo zip y lo almacenará en el archivo zip.hashes. Luego puedes usar John para descifrar el hash.

```
john zip.hashes
```

En los siguientes artículos se habla sobre **John the Ripper**:

- ¿Qué es John the Ripper?
- Crackea contraseñas rápidamente usando John the Ripper
- Password cracking conJohn the Ripper
- Uso práctico de John The Ripper

Se pide:

1. Utilizando **John the Ripper**, **hash-identifier** y el diccionario **rockyou.txt** averigua las contraseñas de los ficheros *texto1.hash, texto2.hash, texto3.hash, texto4.hash y texto5.hash.*
2. En tu máquina Kali Linux crea un usuario y pon una contraseña. Utiliza **John the Ripper** y el diccionario **rockyou.txt** para obtener la contraseña.
3. Utiliza **John the Ripper** y el diccionario **password.lst** para obtener la contraseñas para abrir la carpeta comprimida **.zip**.
4. Utiliza **John the Ripper** y el diccionario **password.lst** para obtener las contraseñas para abrir los documentos **pdf**.

1. Utilizando **John the Ripper**, **hash-identifier** y el diccionario **rockyou.txt** averigua las contraseñas de los ficheros *texto1.hash, texto2.hash, texto3.hash, texto4.hash y texto5.hash.*
    1. Para **comprobar el tipo de hash**, se puede utilizar la herramienta **Hash-identifier**:

       Es una herramienta que le permite reconocer el tipo de hash de una lista de funciones hash conocidas. Dado un hash desconocido, el identificador determina qué función hash probablemente se utilizó para generarlo.

       Hay cientos de algoritmos de hash, la mayoría devuelve un hash como un número, generalmente almacenado en formato hexadecimal. Pero la longitud de este número, algunos caracteres adicionales o el formato de la cadena final permiten reconocer qué tipo de algoritmo se utilizó.

```
┌──(kali㉿kali)-[~/hashes]
└─$ cat texto1.hash
9cecd77a5a7f47c052b8edbe6a7c34dba111c17a288c7bf5ccc8155270366bab
```

```
┌──(kali㉿kali)-[~]
└─$ hash-identifier
   ##########################################################################
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   #                                                                        #
   ##########################################################################

 HASH: 9cecd77a5a7f47c052b8edbe6a7c34dba111c17a288c7bf5ccc8155270366bab
```

Por tanto, utilizando **hash-identifier**, los algoritmos de hash son los siguientes:

prueba.hash algoritmo: SHA256
texto1.hash algoritmo: SHA256
texto2.hash algoritmo: MD5
texto3.hash algoritmo: SHA1
texto4.hash algoritmo: SHA512
texto5.hash algoritmo: MD5

2. **Ejecutar John**:

Para calcular la contraseña del fichero *prueba.hash*, utilizar **John:**

```
john --format=raw-sha256 --wordlist='/home/kali/Escritorio/rockyou.txt' prueba.hash --fork=6
john --show --format=raw-sha256 prueba.hash
```

```
┌──(kali㉿kali)-[~/hashes]
└─$ john --format=raw-sha256 --wordlist='/home/kali/Escritorio/rockyou.txt' prueba.hash --fork=6
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~/hashes]
└─$ john --show --format=raw-sha256 prueba.hash
?:superman

1 password hash cracked, 0 left
```

Para el resto de los archivos:

```
john --format=raw-sha256 --wordlist='/home/kali/Escritorio/rockyou.txt' texto1.hash --fork=6
john --format=raw-md5 --wordlist='/home/kali/Escritorio/rockyou.txt' texto2.hash --fork=6
john --format=raw-sha1 --wordlist='/home/kali/Escritorio/rockyou.txt' texto3.hash --fork=6
john --format=raw-sha512 --wordlist='/home/kali/Escritorio/rockyou.txt' texto4.hash --fork=6
john --format=raw-md5 --wordlist='/home/kali/Escritorio/rockyou.txt' texto5.hash --fork=6
```

2. En tu máquina Kali Linux crea un usuario y pon una contraseña. Utiliza **John the Ripper** y el diccionario *rockyou.txt* para obtener la contraseña.

```
sudo useradd benito
sudo passwd benito
sudo unshadow /etc/passwd /etc/shadow > usuariosypasswords
sudo john --wordlist=/usr/share/john/password.lst usuariosypasswords --format=crypt
sudo john – wordlist=/home/kali/Escritorio/rockyou.txt  usuariosypasswords --format=crypt
sudo john --show usuariosypasswords
```

```
┌──(kali㉿kali)-[~]
└─$ sudo john --wordlist=/home/kali/Escritorio/rockyou.txt  usuariosypasswords --format=crypt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Remaining 1 password hash
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
benito23         (benito)
1g 0:00:35:28 DONE (2024-05-09 04:02) 0.000469g/s 249.0p/s 249.0c/s 249.0C/s benmartin..ben1984
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ sudo john --show usuariosypasswords
kali:kali:1000:1000:,,,:/home/kali:/usr/bin/zsh
benito:benito23:1001:1001::/home/benito:/bin/sh
```

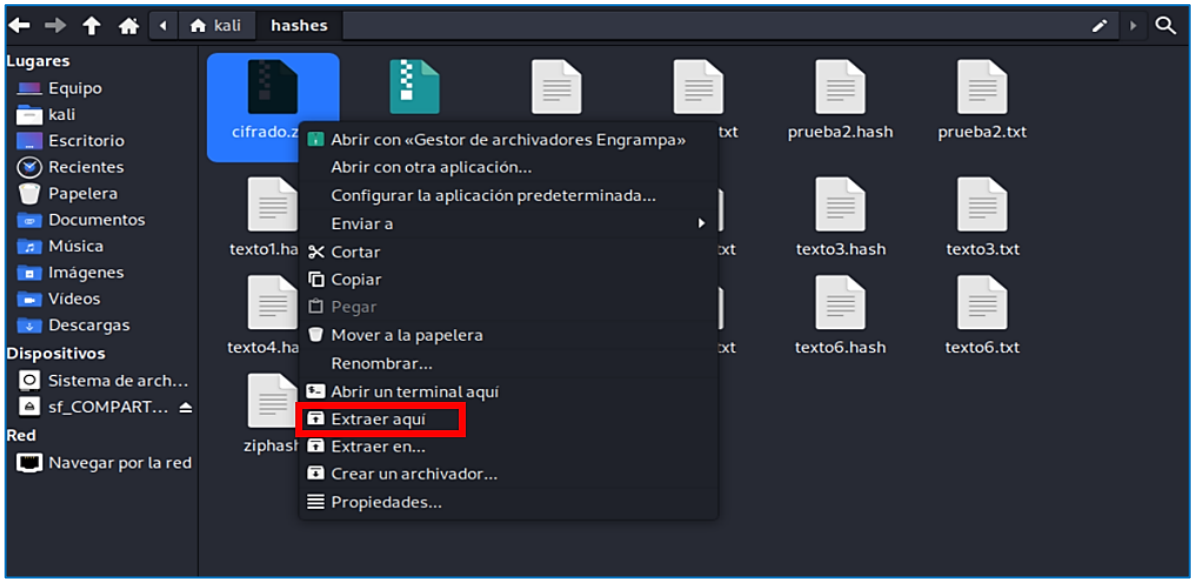
3. Utiliza **John the Ripper** y el diccionario **password.lst** para obtener la contraseña necesaria para abrir la carpeta comprimida **.zip**.

```
zip2john cifrado.zip >ziphash
john --wordlist=/usr/share/john/password.lst ziphash
```

```
┌──(kali㉿kali)-[~/hashes]
└─$ john --wordlist=/usr/share/john/password.lst ziphash
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Loaded hashes with cost 1 (HMAC size) varying from 45440 to 55483
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
harley           (cifrado.zip/IFCD0111_ficha _protected.pdf)
harley           (cifrado.zip/ADGG0508_ficha_protected.pdf)
2g 0:00:00:00 DONE (2024-05-09 04:46) 16.66g/s 29550p/s 59100c/s 59100C/s 123456..sss
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

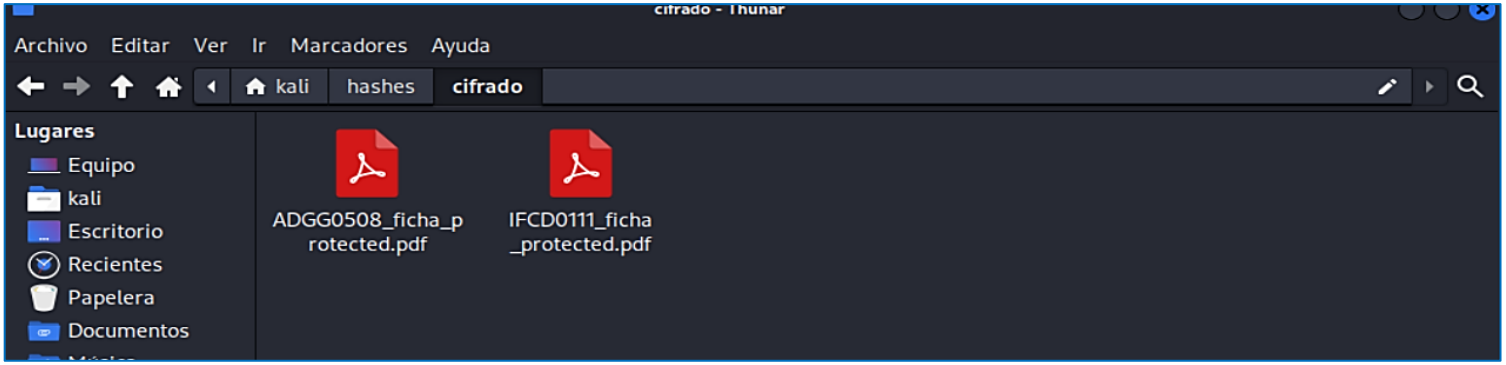
```
┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ john --show ziphash
cifrado.zip/ADGG0508_ficha_protected.pdf:harley:ADGG0508_ficha_protected.pdf:cifrado.zip:cifrado.zip
cifrado.zip/IFCD0111_ficha _protected.pdf:harley:IFCD0111_ficha _protected.pdf:cifrado.zip:cifrado.zip

2 password hashes cracked, 0 left
```

Vamos a abrir la carpeta zip:



Ponemos la contraseña:

Se abre la carpeta comprimida:



4. Utiliza **John the Ripper** y el diccionario **password.lst** para obtener las contraseñas para abrir los documentos **pdf**.

```
/usr/share/john/pdf2john.pl fichero1.pdf >pdfhash1
/usr/share/john/pdf2john.pl fichero2.pdf >pdfhash2
john --wordlist=/usr/share/john/password.lst pdfhash1
john --wordlist=/usr/share/john/password.lst pdfhash2
```

```
┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ /usr/share/john/pdf2john.pl fichero1.pdf >pdfhash1

┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ /usr/share/john/pdf2john.pl fichero2.pdf >pdfhash2

┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ john --wordlist=/usr/share/john/password.lst pdfhash1
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ john --show pdfhash1
fichero1.pdf:secret

1 password hash cracked, 0 left

┌──(kali㉿kali)-[/media/sf_COMPARTIDO/John the ripper]
└─$ john --show pdfhash2
fichero2.pdf:harley

1 password hash cracked, 0 left
```