

Práctica de Laboratorio - Uso de Herramientas de Contraseñas

Objetivos

En esta práctica de laboratorio se cumplirán los siguientes objetivos:

- Parte 1: Investigar los ataques a contraseñas
- Parte 2: Descifrar hashes con ataques de diccionario de hashcat
- Parte 3: Descifrar hashes con John the Ripper mediante diccionario y ataques de fuerza bruta
- Parte 4: Descifrar hashes con RainbowCrack y tablas arcoíris

Aspectos básicos/Situación

Las contraseñas son vulnerables a los ataques. Las contraseñas generalmente se almacenan como hashes cifrados. Un atacante puede capturar los hashes enviados a través de la red mediante herramientas de rastreo o puede obtener acceso a los archivos que contienen hashes de contraseñas en sistemas vulnerables. Cuando el atacante tiene los hashes, puede aplicar ataques de diccionario, tablas arcoíris y fuerza bruta contra ellos fuera de línea para descifrar el hash y recuperar las contraseñas en texto sin formato. Hay muchas herramientas de ataque de contraseñas incluidas con Kali Linux. En esta práctica de laboratorio, se analizarán tres herramientas populares; Hashcat, John the Ripper y RainbowCrack.

Recursos necesarios

- Maquina Virtual Kali (Kali VM) personalizada para el curso de Pirata Ético
- Acceso a Internet

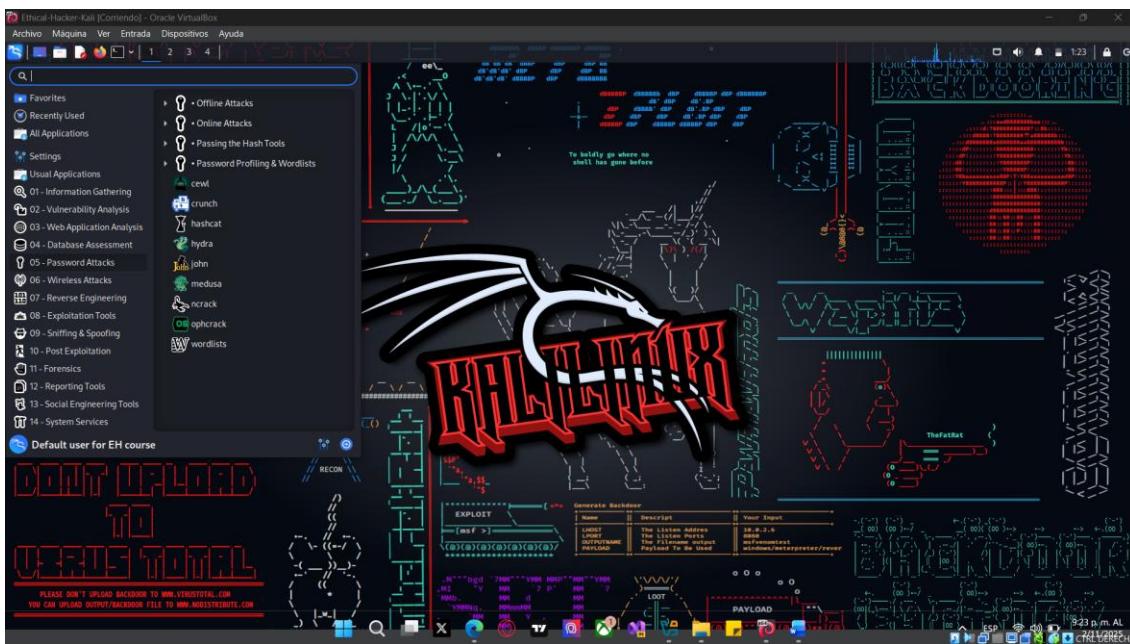
Instrucciones

Parte 1: Investigar ataques de contraseñas

Paso 1: Inicie sesión en Kali Linux y verifique el entorno.

- a. Inicie sesión en Kali usando **kali** como nombre de usuario y contraseña.
- b. Seleccione **Applications > 05 – Password Attacks**.

En el menú Password Attacks de Kali, ¿qué cuatro subcategorías de herramientas de ataque a contraseñas están disponibles?



Ataques sin conexión, ataques en línea, ataques de paso del hash, perfiles de contraseñas y listas de palabras (Offline Attacks, Online Attacks, Passing the Hash Attacks, Password Profiling & Wordlists)

Paso 2: Examine las herramientas de ataque a contraseñas disponibles.

- Haga clic en cada subcategoría de ataque y revise las herramientas de ataque disponibles.
- Pase el cursor sobre cada herramienta. Tenga en cuenta que algunas herramientas tienen un cuadro de texto emergente que contiene una breve descripción de la herramienta. También puede buscar las herramientas en la página Herramientas de Kali para obtener más información sobre ellas y lo que hacen.

¿Qué herramienta es un descifrador de contraseñas de Microsoft que usa tablas arcoíris? ¿Qué subcategoría contiene esta herramienta?

Ophcrack, ataques sin conexión

Parte 2: Descifrar hashes con ataques de diccionario de Hashcat

Paso 1: Cree un archivo que contenga hashes MD5 para descifrar.

Primero, se necesitan algunos hashes de contraseñas MD5. En un ataque real, un atacante ya habrá comprometido un sistema vulnerable para obtener un archivo de contraseñas que contiene hashes de contraseñas almacenados para descifrarlos sin conexión. En este paso, puede simular esto creando un archivo de contraseñas que contiene los hash que descifrará en un próximo paso.

- En una ventana de terminal, cree cinco hashes de destino ingresando los siguientes comandos en el indicador:

```
echo -n 'Password' | md5sum | awk '{ print $1 }' > my_pw_hashes.txt
echo -n 'Password123' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt
echo -n 'Letmein!' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt
echo -n 'ilovedogs' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt
echo -n '1234abcd' | md5sum | awk '{ print $1 }' >> my_pw_hashes.txt
```

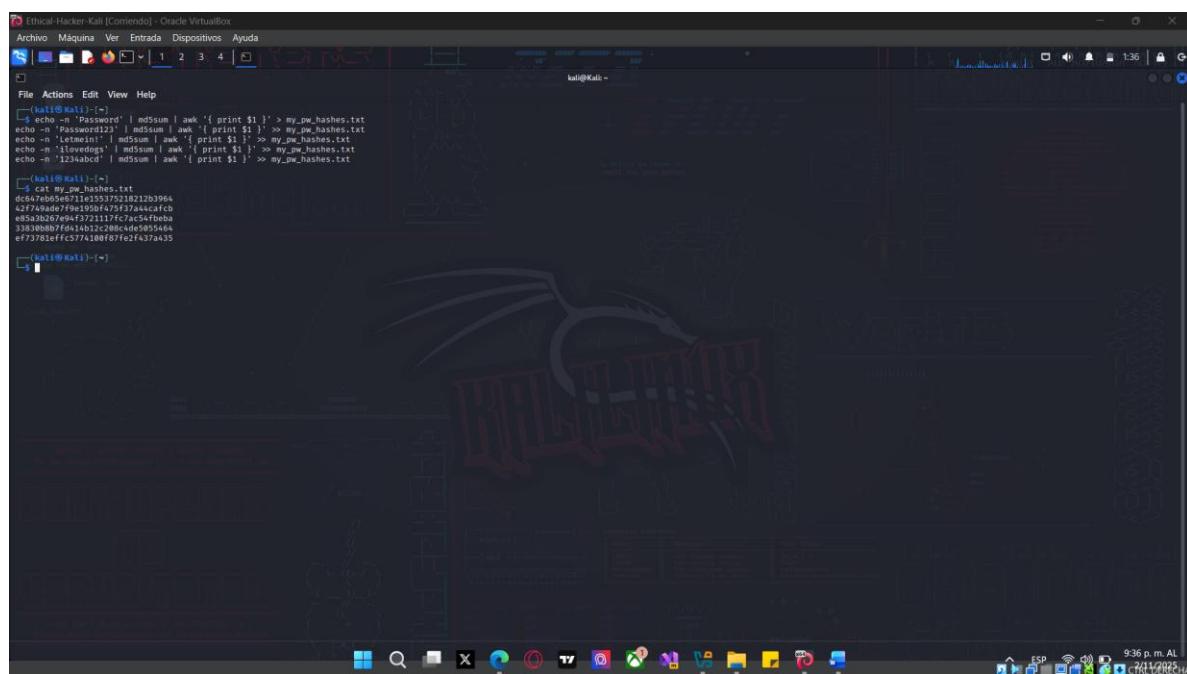
Tenga en cuenta que las contraseñas varían en complejidad.

Los hashes generados se escriben en el archivo **my_pw_hashes.txt**.

- Luego, verifique los hashes de contraseña que acaba de crear ingresando el comando **cat**.

El resultado debería ser similar a este ejemplo:

```
└── (kali㉿Kali)-[~]
└─$ cat my_pw_hashes.txt
dc647eb65e6711e155375218212b3964
42f749ade7f9e195bf475f37a44cafcb
e85a3b267e94f3721117fc7ac54fbbea
33830b8b7fd414b12c208c4de5055464
ef73781effc5774100f87fe2f437a435
```



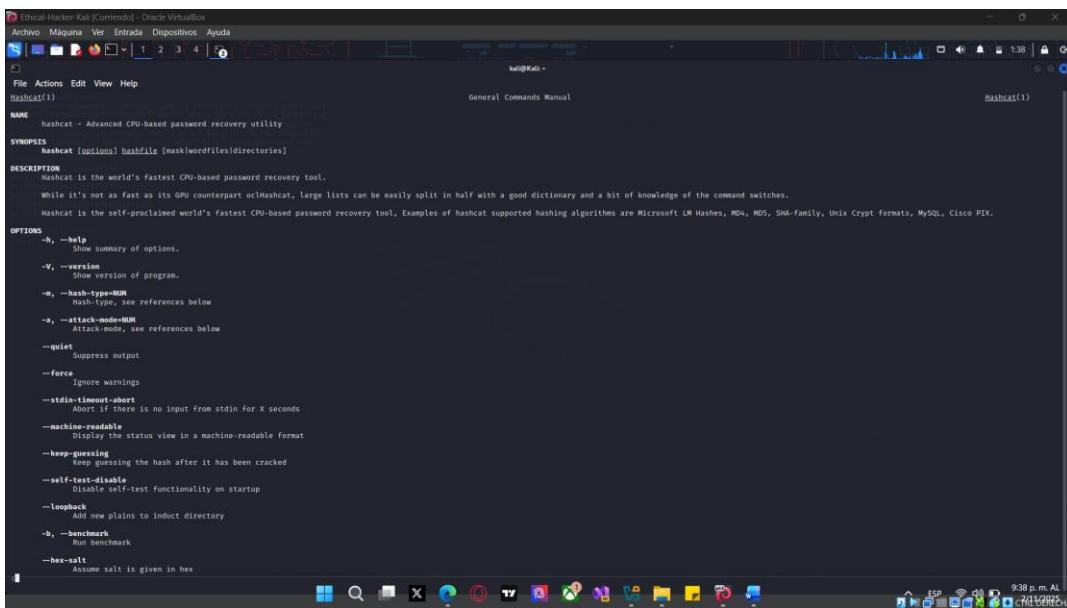
Paso 2: Inicie Hashcat en Kali.

- Abra una nueva consola Kali e ingrese el comando: **man hashcat**.
Esto abre el manual de Hashcat.
- Revise las opciones disponibles en la primera página del manual.

¿Qué se especifica con las opciones **-m** y **-a** ?

La opción -m define el tipo de hash y -a define el modo de ataque.

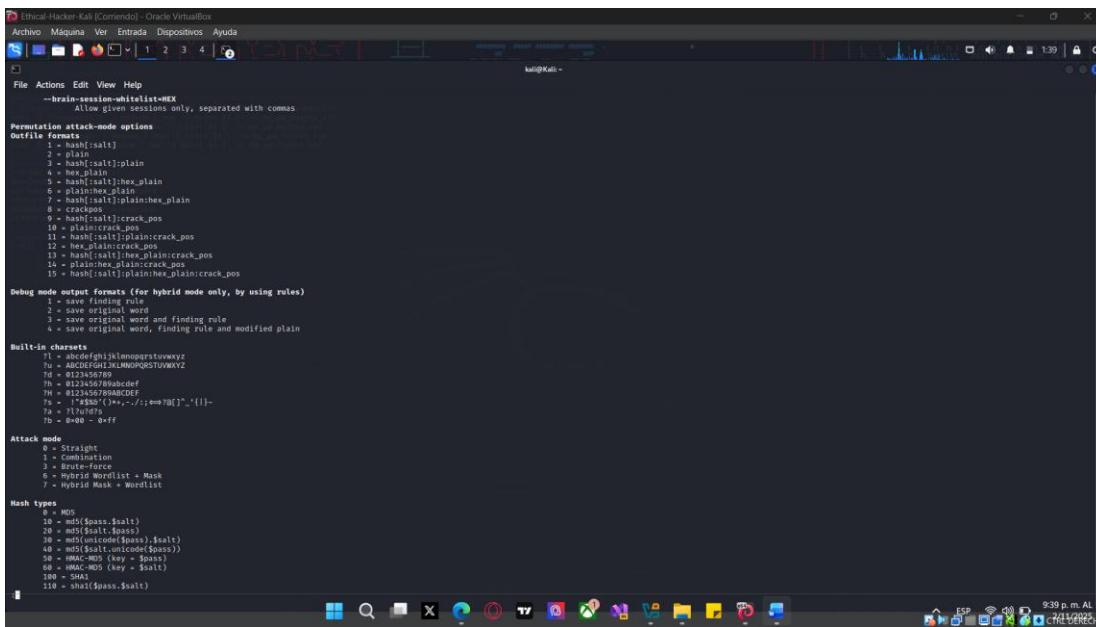
Práctica de Laboratorio - Uso de Herramientas de Contraseñas



- c. Desplácese por la salida de la página del manual para encontrar los valores que se pueden suministrar a cada una de estas opciones.

Utilizará estas opciones pronto en los próximos pasos.

Utilizando las páginas del manual de hashcat, ¿qué tipo de hash y modo de ataque utilizarías para descifrar los hashes de contraseñas en el archivo my_pw_hashes.txt? Explique.



Debido a que los hashes se crearon con md5sum, la opción para los tipos de hash (-m) debe ser 0. En esta instancia, se puede usar el modo de ataque 0 (directo o diccionario).

Paso 3: Vea las listas de palabras disponibles.

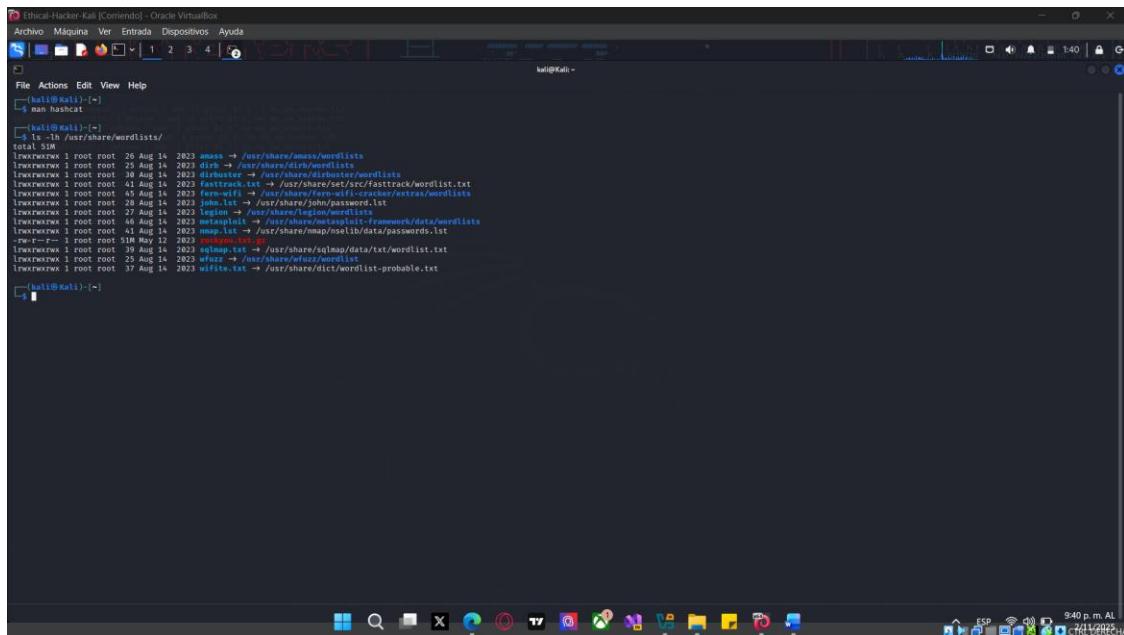
Kali viene con varias listas de palabras integradas. Hashcat necesita usar una lista de palabras para descifrar los valores hash.

- a. Para ver las listas de palabras integradas, ingrese el comando: **ls -lh /usr/share/wordlists/**

```
└── (kali㉿Kali)-[~]
    └─$ ls -lh /usr/share/wordlists/
```

Aquí se enumeran las listas de palabras que se distribuyen con Kali. Utilizaremos la lista de palabras **rockyou.txt**. La lista de palabras rockyou.txt es un diccionario de contraseñas que contiene más de 14 millones de contraseñas.

¿Qué se debe hacer en el archivo rockyou.txt.gz antes de poder usar el archivo de texto de la lista de palabras?



```
terminal-kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
└── man hashcat
(kali㉿kali)-[~]
└─$ ls -lh /usr/share/wordlists/
total 516
lrwxrwxrwx 1 root root 26 Aug 14 2023 answear → /usr/share/mass/wordlists
lrwxrwxrwx 1 root root 25 Aug 14 2023 dirbuster → /usr/share/dir/wordlists
lrwxrwxrwx 1 root root 38 Aug 14 2023 drifuzz → /usr/share/drifuzz/wordlists
lrwxrwxrwx 1 root root 102 Aug 14 2023 fasttrack → /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx 1 root root 45 Aug 14 2023 fcrackazoid → /usr/share/fcrackazoid/wordlists
lrwxrwxrwx 1 root root 28 Aug 14 2023 john → /usr/share/john/passwords.lst
lrwxrwxrwx 1 root root 123 Aug 14 2023 medusa → /usr/share/medusa/wordlists
lrwxrwxrwx 1 root root 46 Aug 14 2023 metasploit → /usr/share/metasploit-framework/data/wordlists
lrwxrwxrwx 1 root root 10 Aug 14 2023 nmap → /usr/share/nmap/nse/lib/data/passwords.lst
lrwxrwxrwx 1 root root 516 May 12 2023 oclhashcat → /usr/share/oclhashcat/wordlists
lrwxrwxrwx 1 root root 39 Aug 14 2023 sqldump → /usr/share/sqlmap/data/txt/wordlist.txt
lrwxrwxrwx 1 root root 25 Aug 14 2023 wifite → /usr/share/wifite/wordlists
lrwxrwxrwx 1 root root 37 Aug 14 2023 wordlist-probable → /usr/share/dict/wordlist-probable.txt
```

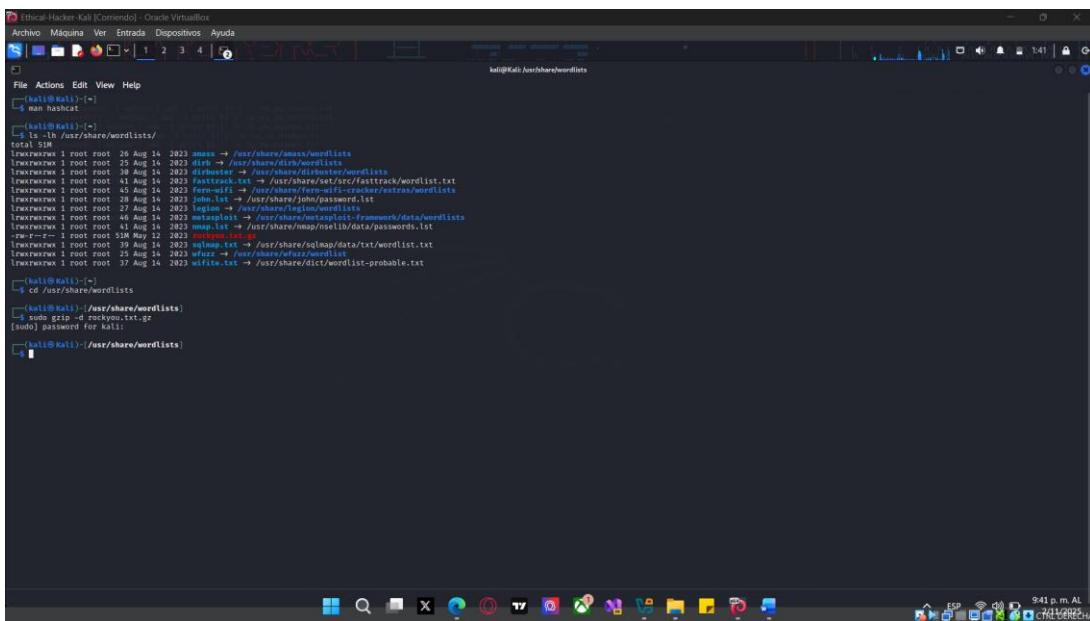
La lista de palabras de rockyou está en un archivo comprimido (indicado por la extensión de archivo .gz). Deberá extraer el archivo de texto del archivo comprimido.

- b. Cambie el directorio a **/usr/share/wordlists** ingresando el comando:

```
└── (kali㉿Kali)-[~]
    └─$ cd /usr/share/wordlists
```

- c. Extraiga el archivo **rockyou.txt.gz** con el comando **gzip**:

```
└── (kali㉿Kali)-[/usr/share/wordlists]
    └─$ sudo gzip -d rockyou.txt.gz
```



```
(kali㉿Kali)-[~]
$ man hashcat
(kali㉿Kali)-[~]
$ ls -lh /usr/share/wordlists/
lswxrwxrwx 1 root root 26 Aug 16 2023 mass → /usr/share/mass/wordlists
lswxrwxrwx 1 root root 25 Aug 16 2023 dirb → /usr/share/dirb/wordlists
lswxrwxrwx 1 root root 25 Aug 16 2023 fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
lswxrwxrwx 1 root root 41 Aug 16 2023 fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
lswxrwxrwx 1 root root 45 Aug 16 2023 Fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
lswxrwxrwx 1 root root 27 Aug 16 2023 metasploit → /usr/share/metasploit-framework/data/wordlists
lswxrwxrwx 1 root root 41 Aug 16 2023 metasploit → /usr/share/metasploit-framework/data/wordlists
-rw-r--r-- 1 root root 51M May 12 2023 rockyou.txt → /usr/share/rockyou/passwords.txt
lswxrwxrwx 1 root root 25 Aug 16 2023 rockyou.txt → /usr/share/rockyou/passwords.txt
lswxrwxrwx 1 root root 25 Aug 16 2023 wifite.txt → /usr/share/dict/wordlist-probable.txt
lswxrwxrwx 1 root root 37 Aug 16 2023 wifite.txt → /usr/share/dict/wordlist-probable.txt

(kali㉿Kali)-[~]
$ cd /usr/share/wordlists
(kali㉿Kali)-[/usr/share/wordlists]
$ sudo gzip -d rockyou.txt.gz
[sudo] password for kali:
(kali㉿Kali)-[/usr/share/wordlists]
```

- d. Enumere el contenido del directorio como se hizo anteriormente con el comando **ls**. Verifique que el archivo **rockyou.txt** ahora esté descomprimido.

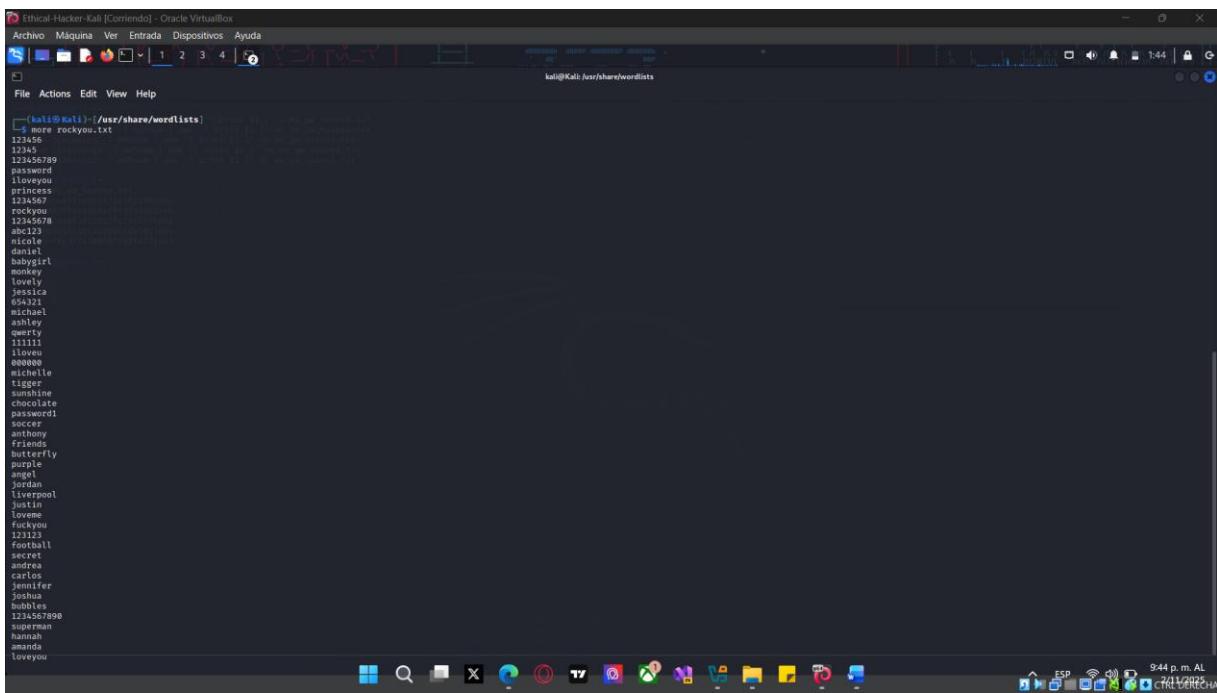
```
└──(kali㉿Kali)-[/usr/share/wordlists]
    └─$ ls
```

- e. Use el comando **more**, seguido del nombre del archivo, para ver el contenido del archivo y ver algunas de las contraseñas que usará hashcat para descifrar sus hashes.

```
└── (kali㉿Kali) - [/ usr / compartir / listas de palabras]
    └─$ más rockyou.txt
```

Las listas de palabras para descifrar hashes o forzar inicios de sesión a menudo se recopilan de volcados de contraseñas que divulgan públicamente información robada de cuentas de usuarios. Desplácese por la salida para tener una idea del contenido del archivo.

¿Cuál parece ser un tipo de contraseña popular? ¿Cómo podría ser útil esta tendencia para un evaluador de penetración?



```
(kali㉿Kali): /usr/share/wordlists
└─$ more rockyou.txt
123456
12345678
password
iloveyou
princess
123456789
rockyou
12345678
alexander
nicole
daniel
babyygirl
merry
lovely
jessica
654321
michael
ashley
queryt
111111
iloveu
000000
michele
tiger
sunshine
chocolate
password1
mister
anthony
friends
marryly
purple
angel
jordan
liverpool
justin
loveme
rockyou
123123
football
secret
secreto
carlos
jennifer
jessus
bubbles
1234567890
superman
annah
amanda
iloveyou
```

Parece haber muchos nombres en la lista. Un evaluador de penetración podría usar las herramientas de OSINT para aprender los nombres de los familiares de los empleados de la empresa. Estos nombres podrían usarse para intentar inicios de sesión y descifrar hashes.

- f. Presione **q** o **Ctrl-z** para salir del contenido del archivo.
- g. Regrese al directorio de inicio.

```
└── (kali㉿Kali) - [/usr/share/wordlists]
    └─$ cd /home/kali
```

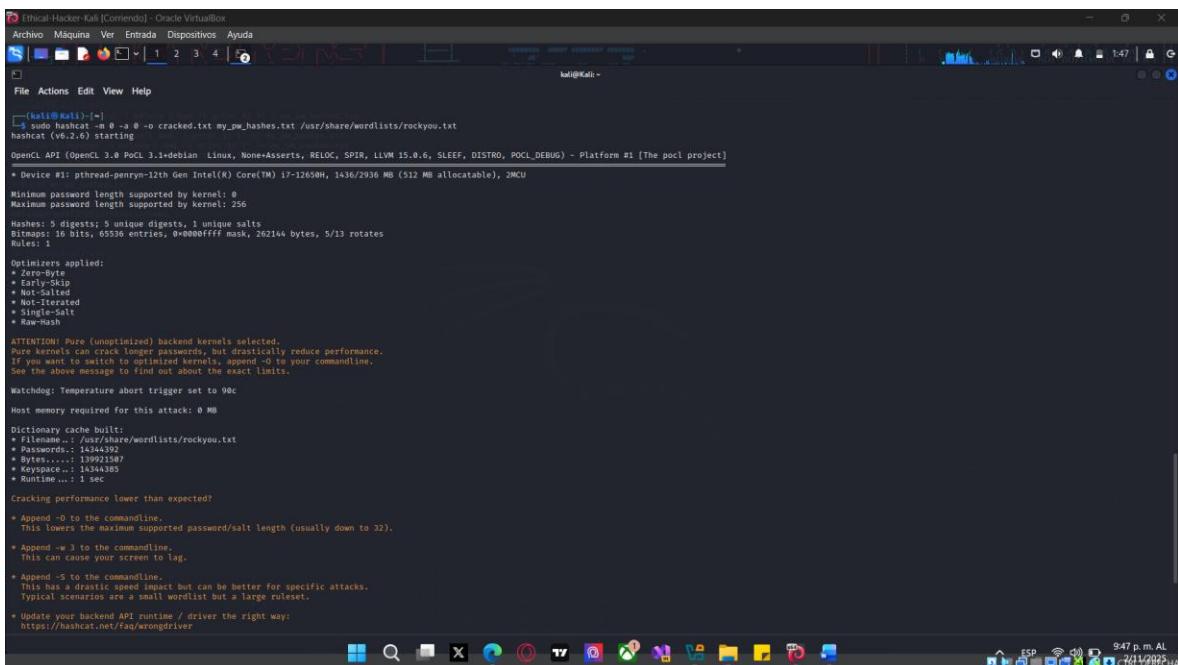
Paso 4: Descifre hashes con Hashcat.

- a. Para descifrar los hashes contenidos en el archivo **my_pw_hashes.txt**, use el siguiente comando:

```
└── (kali㉿Kali) - [~]
    └─$ sudo hashcat -m 0 -a 0 -o cracked.txt my_pw_hashes.txt
        /usr/share/wordlists/rockyou.txt
```

Este comando genera las contraseñas descifradas en el nuevo archivo **cracked.txt**.

Práctica de Laboratorio - Uso de Herramientas de Contraseñas



```
(kali㉿Kali)-[~]
└─$ sudo hashcat -m 0 -o cracked.txt my_pw_hashes.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1-debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: pthread-penryn-12th Gen Intel(R) Core(TM) i7-12650H, 1436/2936 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 5 digests; 5 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte-Drop
* Early-Stop
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Mask

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c
Host memory required for this attack: 0 MB

Dictionary cache built:
* Words...: /usr/share/wordlists/rockyou.txt
* Passwords.: 16344392
* Bytes....: 139921587
* Keyspace..: 14344385
* Runtime ...: 1 sec

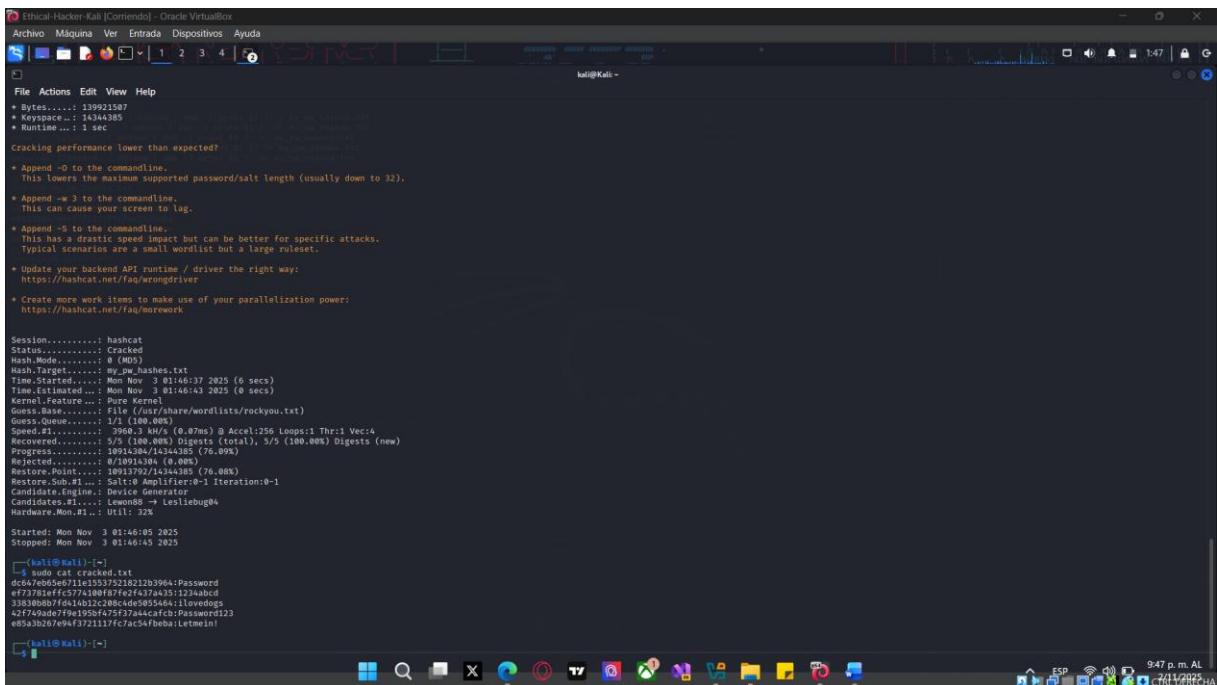
Cracking performance lower than expected?
* Append -O to the commandline.
  This lowers the maximum supported password/salt length (usually down to 32).
* Append -w 3 to the commandline.
  This can cause your screen to lag.
* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.
* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

(kali㉿Kali)-[~]
```

- b. Para ver el contenido del archivo cracked.txt y las contraseñas de texto sin formato, ingrese el comando:

```
(kali㉿Kali)-[~]
└─$ sudo cat cracked.txt
```

¿Cuántas contraseñas se decodificaron?



```
(kali㉿Kali)-[~]
└─$ sudo cat cracked.txt
my_pw_hashes.txt:Password
e73781ffcc5774180000000000000000:Password
33333333333333333333333333333333:Password
42f7494ade7f9e195bf175f37aa4acafcb:Password123
eb5a3b267e94f3721117fc7ac54fbaba:Letmein123

(kali㉿Kali)-[~]
```

Las respuestas pueden variar, pero Hashcat debería descifrar rápidamente las cinco.

Parte 3: Descifrar hashes con John the Ripper mediante diccionario y ataques de fuerza bruta

Paso 1: Vea el archivo de ayuda de John the Ripper.

En una ventana de terminal, introduzca el comando: **john -h** para ver el archivo de ayuda de John the Ripper.

```
(kali㉿Kali)-[~]
└$ john -h
[...]
File Actions Edit View Help
(hal@kali)-[~]
└$ john -h
[...]
Usage: john [OPTIONS] [PASSWORD-FILES]

--help          Print usage summary
--single[=SECTION[,...]]  "Single crack" mode, using default or named rules
--single=rules[,...]    Same, using named rules
--single=seed[WORD]     All salts in single mode
--single=wordlist=FILE *Short wordlist with static seed words/morphemes
--single=user-seed=FILE Wordlist with seeds per username (user:password[s])
--single-pair-max=N   Override max. number of word pairs generated (6)
--no-single-pair      Disable single word pair generation
--no-single-retest-guess  Override config for SingleRetestGuess
--wordlist[=FILE]      stdin: wordlist from stdio; file: from FILE or stdin
--wordlist=FILE        pipe: like stdin, but respects wordlist; allows rules
--rules[=SECTION[,...]] Enable word mangling rules (for wordlist or PRINCE
modes), using default or named rules
--rules=:rule[,...]     Same, using immediate" rule(s)
--rules=stack[=SECTION[,...]] Stacked rules, applied after regular rules or to
modes that otherwise don't support rules
--rules=stack=rule[,...] Same, using "immediate" rule(s)
--ruleskip[=FILE]      Like wordlist, but extract words from a .pot file
--mem-file-size=SIZE   Size threshold for wordlist preload (default 2048 MB)
--dupe-suppression    Suppress all dupes in wordlist (and force preload)
--incremental=INCREMENT  Turn incremental attack into increment MODE
--incremental=charcount=N  Override CharCount for incremental mode
--external=MODE        External mode or word filter
--mask[=MASK]          Mask mode uses MASK (or default from john.conf)
--maskfile[=FILE]       Mask file, see doc/MASK
--mkv-stats=FILE        "Markov" stats file
--prince[=FILE]         PRINCE mode, read words from FILE
--prince-loopback[=FILE] Fetch words from a .pot file
--prince-elem-cnt-min=N Minimum number of elements per chain (1)
--prince-elem-cnt-max=[-]N Maximum number of elements per chain (negative N is
relative to word length) (8)
--prince-skip=N        Ignore skip
--prince-limit=N       Limit number of candidates generated
--prince-wl-dist=LEN   Calculate length distribution from wordlist
--prince-wl-max=N      Load only N words from input wordlist
--prince-case-permute  Permute case first letter
--prince-casefile=FILE  Name of case profile (not available with case permute)
--prince-keyspace      Just show total keyspace that would be produced
(disregarding skip and limit)
--subsets[=<CHARSET>]  Subsets (see doc/SUBSETS)
--subsets-required=N   The N first characters of "subsets" charset are
the "required set"
--subsets-min-diff=N   Minimum unique characters in subset
--subsets-max-diff=[-]N Maximum unique characters in subset (negative N is
relative to word length) (8)
```

Paso 2: Descifre los valores hash con John the Ripper.

Utilice el siguiente comando para descifrar los hash en el archivo **my_pw_hashes**. Esto puede llevar algún tiempo.

```
(kali㉿Kali)-[~]
└$ john --format=raw-md5 my_pw_hashes.txt
```

John muestra las contraseñas descifradas en naranja. ¿Qué contraseñas se descifraron con la lista de palabras de contraseñas?

```
(ethical-Hacker-Kali [Corriendo] - Oracle VirtualBox)
File Actions Edit View Help
--node-MIN[=MAX]TOTAL
--env-memory-LEVEL
--log-stems
--verbosity-N
--no-log
--max-always-valid=N
--catch-up=NAME
--configFILE
--encoding=NAME
--input-encoding=NAME
--internal-codepage=NAME
--internal-encoding=NAME
--force-tty
--field-separator=CHAR
--no-keep-guessing
--list=MAT
--length-N
--min-length-N
--max-length-N
--max-candidates=[-]N
--max-run-time=[-]N
--skip=N
--no-loader-dupecheck
--not=NAME
--reject-salts=N
--reject-printable
--tunis=HOW
--subformat=FORMAT
--format=[NAME|CLASS][,...]
john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 my_pw_hashes.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MDS 128/128 SSE2 4x3])
Warning: no OpenMP support, consider changing hash type, consider -fork=2
Progress: 0:00:03:19 3/3 0.01504g/s 24133Kc/s 48410Kc/s 1b3d080..1b3d08br
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Progressing with wordlist:/usr/share/john.passwords
Password? (7)
Letmein! (7)
Processing with incremental:ASCII
1b3d080..1b3d08br
3g 0:00:03:19 3/3 0.01504g/s 24133Kc/s 48410Kc/s 1b3d080..1b3d08br
3g 0:00:03:24 3/3 0.01574g/s 24133Kc/s 24133Kc/s duolito123
3g 0:00:03:24 3/3 0.01574g/s 24133Kc/s 483810Kc/s 483810Kc/s 23askhbj..23askhbj
3g 0:00:03:25 3/3 0.01462g/s 24078Kc/s 24078Kc/s signam15..signam15
9:52 p.m. AL
CTRE LUECRECHIA
```

Password, y Letmein! Deben descifrarse de inmediato, ya que están en las listas de palabras utilizadas por John.

En este caso, John usa una lista mínima de palabras de contraseña de manera predeterminada para descifrar rápidamente las contraseñas comunes.

¿Qué hace John the Ripper si hay valores hash que no puede descifrar con sus listas de palabras?

John the Ripper cambia a estrategias incrementales (fuerza bruta) en los hashes restantes.

Si deja que John continúe ejecutándose el tiempo suficiente, eventualmente descifrará las contraseñas restantes. (Tenga en cuenta que esto puede demorar entre 10 y 20 minutos). Presione **Ctrl-C** para cancelar en cualquier momento después de haber descifrado algunas contraseñas, si lo desea.

Paso 3: Utilice listas de palabras más grandes.

La lista de palabras predeterminada para John the Ripper es bastante pequeña. John puede usar otras listas de palabras, como la lista de palabras rockyou.txt. También es posible descargar listas de palabras adicionales de Internet.

Utilice el siguiente comando para indicarle a John the Ripper que utilice la lista de palabras rockyou.txt.

```
(kali㉿Kali)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5
my_pw_hashes.txt
```

Paso 4: Utilice la fuerza bruta.

Para indicar a John the Ripper que use solo el descifrado por fuerza bruta, use el siguiente comando:

Tenga en cuenta que el uso de la fuerza bruta puede llevar mucho tiempo para descifrar los hash de contraseña. Una GPU potente puede tardar muchas horas en descifrar una contraseña compleja de 8 caracteres.

MD5 se considera demasiado débil para usar. Sin embargo, observe cuánto tiempo lleva descifrar incluso un hash MD5 mediante la fuerza bruta. Cancele el proceso con **Ctrl-C** o **q**.

Paso 5: Muestre sus contraseñas descifradas.

En este ejemplo, si interrumpió el proceso de descifrado de contraseñas con john y el formato RAW-MD5, aún puede revisar las contraseñas descifradas con la opción **--show** .

```
(kali㉿Kali)-[~]
└$ john --show --format=raw-md5 my_pw_hashes.txt

[+] Ethical-Hacker (Comando) - Oracle VirtualBox
 Archivo Máquina Ver Entrada Dispositivos Ayuda
File Actions Edit View Help
john@Kali:~$ john --show --format=raw-md5 my_pw_hashes.txt
Warning: detected hash type "1B", but the string is also recognized as "dynamic-md5($p)"
Use the "--format=dynamic-md5($p)" option to force loading these as that type instead
Warning: detected hash type "IM", but the string is also recognized as "MDA5-128-4"
Use the "--format=MDA5-128-4" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "W02"
Use the "--format=W02" option to force loading these as that type instead
Warning: detected hash type "MD5", but the string is also recognized as "md5"
Use the "--format=md5" option to force loading these as that type instead
Warning: detected hash type "IM", but the string is also recognized as "mcash"
Use the "--format=mcash" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "encash"
Use the "--format=encash" option to force loading these as that type instead
Warning: detected hash type "1B", but the string is also recognized as "Raw-MD4"
Use the "--format=Raw-MD4" option to force loading these as that type instead
Warning: detected hash type "IM", but the string is also recognized as "Raw-MD4"
Use the "--format=Raw-MD4" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"
Use the "--format=Raw-MD5" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"
Use the "--format=Raw-MD5" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-SHA1-AcCrypt"
Use the "--format=Raw-SHA1-AcCrypt" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-SHA1-128"
Use the "--format=Raw-SHA1-128" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Snefru-128"
Use the "--format=Snefru-128" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "ZipMonster"
Use the "--format=ZipMonster" option to force loading these as that type instead
Using default target encoding: CP950
Loaded 10 password hashes with no different salts (LM [DES 128/128 55E2])
Warning: detected hash type "IM", but the string is also recognized as "Raw-MD5"
For better crackability for this hash type, consider --fork2
Will run 7 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Session aborted

[kali㉿Kali]-[~]
└$ john --show --format=raw-md5 my_pw_hashes.txt
?Password
?Password123
?1etadmin
?password123
?doggs
?1234abc

5 password hashes cracked, 0 left
```

Paso 6: Experimentar

Si tiene tiempo, pruebe algunas contraseñas complejas de diferentes longitudes de 4 a 8 caracteres. Intente descifrar por fuerza bruta las contraseñas con John the Ripper en modo incremental. Cree un archivo que contenga hashes de contraseñas y luego ejecute la herramienta.

Parte 4: Descifrar hashes con RainbowCrack y tablas arcoiris

Nota: RainbowCrack no está disponible en la VM que utiliza CPU ARM (Apple M1/M2).

Paso 1: Instale RainbowCrack.

Es posible que deba instalar la utilidad RainbowCrack. Rainbow Crack se diferencia de las utilidades de descifrado de hashes que utilizan algoritmos de fuerza bruta en que utiliza tablas de arco iris para descifrar hashes de contraseñas.

Para instalar RainbowCrack ingrese el siguiente comando:

```
| └─(kali㉿Kali)-[~]  
└─$ sudo apt install rainbowcrack
```

```
Ethical-Hacker-Kali [Comiendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
[ kali@Kali ~ ]$ john --show --format=raw-md5 my_pw_hashes.txt
?Password
?1234567890
?password123
?letmein
?ilovetodo
?1234567890
?1234567890

5 password hashes cracked, 0 left

[ kali@Kali ~ ]$ sudo apt install rainbowcrack
Reading package lists... done
Building dependency tree... done
Reading state information... done
The following NEW packages will be installed:
rainbowcrack
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 130 kB of archives.
After this operation, 586 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali/main/rainbowcrack/amd64 rainbowcrack_amd64_1.0-0kali1 [48 kB]
404  Not Found [IP: 54.39.128.230 80]
Failed to Fetch http://http.kali.org/kali/main/rainbowcrack/rainbowcrack_1.0-0kali1_amd64.deb 404  Not Found [IP: 54.39.128.230 80]
Unable to fetch some archives; maybe run apt-get update or try with --fix-missing

[ kali@Kali ~ ]$
```

Paso 2: Creación de tablas de arcoíris con rtgen.

Las tablas de Rainbow son archivos comunes que pueden crearse con RainbowCrack o descargarse de Internet. La creación de una tabla de arcoíris puede llevar una cantidad considerable de tiempo y espacio de almacenamiento, ya que son muy grandes, con un tamaño que va desde los 20 GB hasta más de un terabyte.

- a. Cree una pequeña tabla de arcoíris simple que descifre contraseñas MD5 de hasta 3 caracteres con solo letras minúsculas.

El programa **rtgen** se usa para generar tablas de arcoíris basadas en parámetros especificados por el usuario.

- 1) Ingrese el comando **rtgen -h** y revise las opciones.

Las tablas de arcoíris de ejemplo se proporcionan en la parte inferior del resultado.

```
Ethical-Hacker-Kali [Corriendo] - Oracle VirtualBox
Archivo Maquina Ver Entrada Dispositivos Ayuda
[ 1 2 3 4 ] kalin@kali: ~

File Actions Edit View Help
Use the "--format=crashdump" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP950
Using default memory size: 1024 MB
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Press Ctrl+C twice to abort, almost any other key for status
Session aborted

[kalin@kali: ~]
$ john --show --format=raw-md5 my_pw_hashes.txt
?@password
?@password123
?@letmein!
?@l0rdedog
?#@24hacked

5 password hashes cracked, 0 left

[kalin@kali: ~]
└─$ sudo apt-get install rainbowcrack
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
rainbowcrack
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 139 kB of archives.
After this operation, 596 kB of additional disk space will be used.
Err: http://http.kali.org/kali/main amd64 rainbowcrack amd64 1.8-0kali1
404  Not Found [IP: 54.39.128.230 80]
Failed to fetch http://http.kali.org/kali/main/rainbowcrack_1.8-0kali1_amd64.deb 404  Not Found [IP: 54.39.128.230 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?

[kalin@kali: ~]
└─$ cigen -h
Command "rgen" not found, but can be installed with:
sudo apt install rgen
Do you want to install it? (y/N)y
sudo apt install rainbowcrack
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
rainbowcrack
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 139 kB of archives.
After this operation, 596 kB of additional disk space will be used.
Err: http://http.kali.org/kali/main amd64 rainbowcrack amd64 1.8-0kali1
404  Not Found [IP: 54.39.128.230 80]
Failed to fetch http://http.kali.org/kali/main/rainbowcrack_1.8-0kali1_amd64.deb 404  Not Found [IP: 54.39.128.230 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?

[kalin@kali: ~]
```

- 2) Cree una tabla de arcoíris ingresando:

```
└── (kali㉿Kali)-[~]
└─$ sudo rtgen md5 loweralpha 1 3 0 1000 1000 0
```

```
(kali㉿Kali)-[~]
└─$ sudo rtgen md5 loweralpha 1 3 0 1000 1000 0
rainbow table md5_loweralpha1-3_0_1000x1000_0.rt parameters
hash algorithm: md5
hash length: 16
charset name: loweralpha
charset digest: abcdefghijklmnopqrstuvwxyz
digest data in hex: 03 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a
charset length: 26
plaintext length range: 1 - < 1000000000
seed length: 1000000000
plaintext total: 18278
sequential starting point begin from 0 (0x0000000000000000)
generating...
1000 of 1000 rainbow chains generated (0 m 0.1 s)
```

Este comando crea una tabla de arcoíris que puede descifrar contraseñas de tres caracteres y solo letras minúsculas. La aplicación creó un archivo con 1000 entradas. La creación de tablas de arcoíris más complejas puede llevar mucho tiempo y consumir muchos recursos.

- b. Verifique que se cree la tabla del arcoíris. Muestre el contenido del directorio rainbowcrack ingresando el comando:

```
└── (kali㉿Kali)-[~]
└─$ cd /usr/share/rainbowcrack
└── (kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ ls
```

```
Ethical-Hacker-Kali [Corriendo] - Oracle VirtualBox
Archivo Maquina Ver Entrada Dispositivos Ayuda
File Actions Edit View Help
[+] kali@Kali:[~]
└─# ./rtgen md5 lowerlalpha 1 3 @ 1000 1000 #
rainbow table md5_lowerlpha=1_3_0_1000x1000_0_rt parameters
hash algorithm: md5
hash length: 16
charset name: lowerlpha
charset data: abcdefghijklmnopqrstuvwxyz
charset data in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a
block size: 20
plaintext length range: 1 ~ 3
reduce offset: 0x00000000
plaintext total: 18278

sequential starting point begin from 0 (0x0000000000000000)
generating...
1000 of 1000 rainbow chains generated (@ m 0.1 s)

[+] kali@Kali:[~]
└─# cd /usr/share/rainbowcrack
[+] kali@Kali:[/usr/share/rainbowcrack]
└─# ./rclibc.so charset.txt md5_lowerlpha=1_3_0_1000x1000_0_rt rcrack readme.txt rt2rtc rtc2rt rtmrgn rtmrgs
[+] kali@Kali:[/usr/share/rainbowcrack]
```

La tabla de arcoíris recién creada debe estar en el directorio como un archivo .rt.

Paso 3: Ordene la tabla del arcoíris.

- a. A continuación, se debe ordenar la tabla del arcoíris. Ingresando el comando : **`sudo rtsort`**. en el indicador. (**Nota:** asegúrese de incluir el espacio y el período después de **`rtsort`** como parte del comando)

```
└─(kali㉿Kali)-[/usr/share/rainbowcrack]
└$ sudo rtsort .
```

- b. Genere un hash para una contraseña simple de 3 caracteres que luego pueda descifrarse. Ingrese el comando: `echo -n 'dog' | md5sum | awk '{print $1}'`.

```
└─(kali㉿Kali)-[/usr/share/rainbowcrack]
└$ echo -n 'dog' | md5sum | awk '{print $1}'
06d80eb0c50b49a509b49f2424e8c805
```

- c. Descifra el hash con la tabla arcoíris usando RainbowCrack. En el indicador, ingrese el comando **rcrack -h 06d80eb0c50b49a509b492424e8c805**.

```
└─(kali㉿Kali)-[/usr/share/rainbowcrack]
└─\$ rcrack . -h 06d80eb0c50b49a509b492424e8c805
```

En milisegundos, RainbowCrack debería descifrar el hash y revelar la contraseña **dog**.

```

ethical-hacker-kali [Comiendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
File Actions Edit View Help
└─(kali㉿kali)-[~]
  └─$ sudo rgen md5 loweralpha 1 3 0 1000 1000 0
  rainbow table md5_loweralpha#1-3_0_1000-1000_0.rt parameters
    Minimum length: 1
    Hash length: 16
    Charset name: loweralpha
    Charset data: abcdefghijklmnopqrstuvwxyz
    Charset data in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a
    Charset length: 26
    Plaintext length range: 1 - 3
    Recurse offset: +0x00000000
    Plaintext total: 18278
  sequential starting point begin from 0 (@=0x0000000000000000)
  generating ...
  1000 of 1000 rainbow chains generated (0 = 0.1 s)

(kali㉿kali)-[~]
└─$ cd /usr/share/rainbowcrack
(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ ls
alglib.so charset.txt md5_loweralpha#1-3_0_1000-1000_0.rt rcrack readme.txt rt2rtc rtc2rt rgen rmerge rsort
(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ sudo rsort
rainbowcrack 0.8
Copyright 2020 RainbowCrack Project. All rights reserved.
http://project-rainbowcrack.com/
usage: ./rtsort path

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ echo -n dog | md5sum | awk '{print $1}'
06d80bebc050a9509a492424e8c805

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ rcrack . -h 06d80bebc050a9509a492424e8c805
invalid hash 06d80bebc050a9509a492424e8c805

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ rcrack . -h 06d80bebc050a9509a492424e8c805
./md5_loweralpha#1-3_0_1000-1000_0.rt is not sorted
result

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ 

```

- d. También puede descifrar los hash contenidos en un archivo .txt, como se hizo en la parte 1 de la práctica de laboratorio. Para crear un archivo .txt con algunos hash, ingrese los siguientes comandos en el indicador:

```

echo -n 'fox' | md5sum | awk '{print $1}' > ~/my_rainbow_hashes.txt
echo -n 'boo' | md5sum | awk '{print $1}' >> ~/my_rainbow_hashes.txt
echo -n 'pop' | md5sum | awk '{print $1}' >> ~/my_rainbow_hashes.txt

```

Para descifrar los hashes en el archivo, ingrese el comando **rcrack . -l ~/my_rainbow_hashes.txt** en el indicador. La opción **-l** le dice a rcrack que use un archivo de lista hash como entrada.

```

└─(kali㉿Kali)-[/usr/share/rainbowcrack]
└─$ rcrack . -l ~/my_rainbow_hashes.txt

```

```

ethical-hacker-kali [Comiendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
File Actions Edit View Help
└─(kali㉿kali)-[~]
  └─$ cd /usr/share/rainbowcrack
  (kali㉿kali)-[~/usr/share/rainbowcrack]
  alglib.so charset.txt md5_loweralpha#1-3_0_1000-1000_0.rt rcrack readme.txt rt2rtc rtc2rt rgen rmerge rsort
  (kali㉿kali)-[~/usr/share/rainbowcrack]
  └─$ sudo rsort
rainbowcrack 0.8
Copyright 2020 RainbowCrack Project. All rights reserved.
http://project-rainbowcrack.com/
usage: ./rtsort path

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ echo -n dog | md5sum | awk '{print $1}'
06d80bebc050a9509a492424e8c805

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ rcrack . -h 06d80bebc050a9509a492424e8c805
invalid hash 06d80bebc050a9509a492424e8c805

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ rcrack . -h 06d80bebc050a9509a492424e8c805
./md5_loweralpha#1-3_0_1000-1000_0.rt is not sorted
result

(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ echo -n fox | md5sum | awk '{print $1}' > ~/my_rainbow_hashes.txt
echo -n boo | md5sum | awk '{print $1}' >> ~/my_rainbow_hashes.txt
echo -n pop | md5sum | awk '{print $1}' >> ~/my_rainbow_hashes.txt
(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ rcrack . -l ~/my_rainbow_hashes.txt
./md5_loweralpha#1-3_0_1000-1000_0.rt is not sorted
result

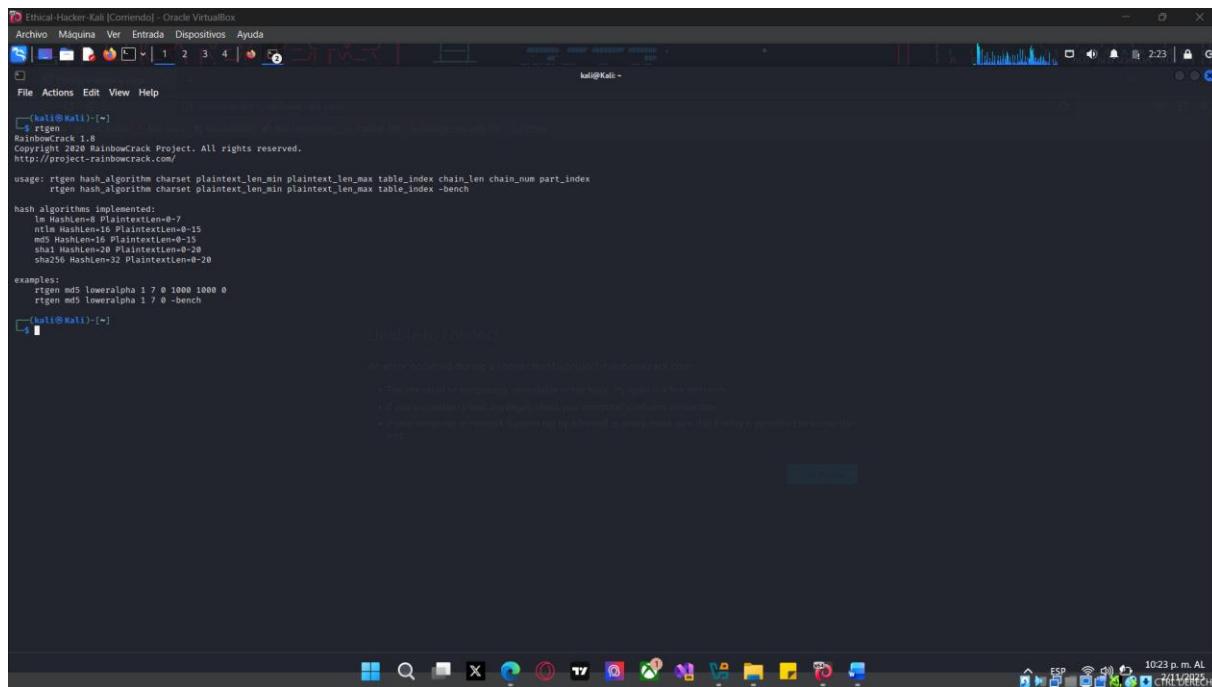
(kali㉿kali)-[~/usr/share/rainbowcrack]
└─$ 

```

Paso 4: Explore recursos para descargar tablas de arcoíris.

Además de generar tablas arcoiris con el comando **rtgen**, hay muchos recursos en Internet para descargar tablas arcoiris.

Abra un navegador web y busque **descarga de tablas arcoíris**.



The screenshot shows a terminal window titled 'Ethical-Hacker-Kali [Corriendo] - Oracle VirtualBox'. The terminal displays the usage information for the 'rtgen' command, which is part of the RainbowCrack project. It lists various hash algorithms and their plaintext lengths (e.g., md5 HashLen=8, ntlm HashLen=16, sha1 HashLen=20, sha256 HashLen=32). Examples of command-line usage are shown, such as 'rtgen md5 loweralpha 1 7 0 1000 1000 0' and 'rtgen md5 loweralpha 1 7 0 0 -bench'. The terminal also shows a message from the RainbowCrack project's website: 'An effort is being made to keep this tool up-to-date. If you find any bugs or have any suggestions, please feel free to contact us at project@rainbowcrack.com'. The desktop environment includes icons for various applications like a browser, file manager, and terminal, along with system status indicators like battery level and signal strength.

```
<kali㉿Kali:~>
rtgen
RainbowCrack 1.8
Copyright 2020 RainbowCrack Project. All rights reserved.
http://project-rainbowcrack.com/
usage: rgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index
      rgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index -bench

hash algorithms implemented:
  md5 HashLen=8 PlaintextLen=0-7
  ntlm HashLen=16 PlaintextLen=0-15
  md5 HashLen=16 PlaintextLen=0-20
  sha1 HashLen=20 PlaintextLen=0-20
  sha256 HashLen=32 PlaintextLen=0-20

examples:
  rgen md5 loweralpha 1 7 0 1000 1000 0
  rgen md5 loweralpha 1 7 0 0 -bench
<kali㉿Kali:~>
```

Preguntas de reflexión

1. ¿Por qué la complejidad y la longitud son tan importantes al crear contraseñas?

Las respuestas pueden variar, pero, como ilustra la práctica de laboratorio, los hashes de contraseñas simples y breves se descifran casi de inmediato mediante los diccionarios y las tablas de arcoíris. Incluso las contraseñas bastante complejas se pueden descifrar en cuestión de horas.

2. Además de la complejidad y la longitud, ¿qué otras medidas se pueden tomar para proteger las contraseñas?

Las respuestas pueden variar, pero cambiar las contraseñas periódicamente, proteger los servidores que alojan los archivos de cuentas de usuario y contraseñas, proteger las redes cableadas e inalámbricas para que los atacantes no puedan capturar hashes de contraseñas en tránsito.