

Python para Hackers



Python para Hackers

Contenido del curso

Capítulo 1. Introducción

Capítulo 2. Primeros pasos

Capítulo 3. Python

Capítulo 4. Hands-On

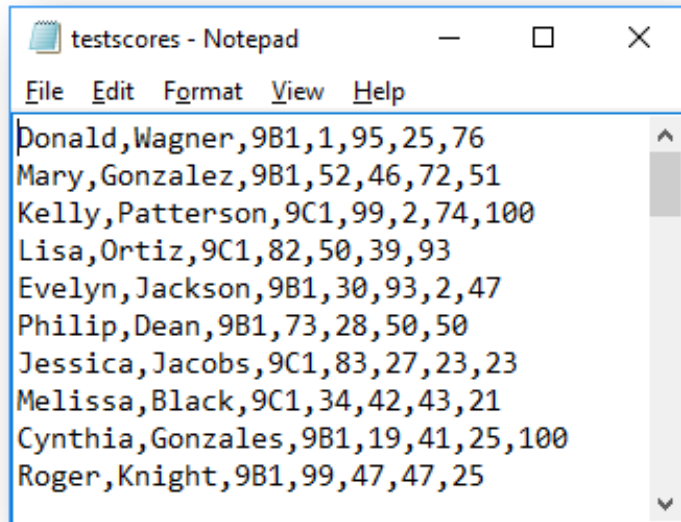
Tema de hoy

Ataques de diccionario | Banner Grabbing | Reconocimiento de máquinas | Servidor/Cliente TCP | Web Server | Web Scraping | Packet Sniffing con Scapy | Paramiko (Cliente SSH) | Nmap con Python | MacChanger | Fuerza bruta de Directorios Web | Fuerza bruta a formularios de autenticación (web)



Archivos

Archivos



```
testscores - Notepad
File Edit Format View Help
Donald,Wagner,9B1,1,95,25,76
Mary,Gonzalez,9B1,52,46,72,51
Kelly,Patterson,9C1,99,2,74,100
Lisa,Ortiz,9C1,82,50,39,93
Evelyn,Jackson,9B1,30,93,2,47
Philip,Dean,9B1,73,28,50,50
Jessica,Jacobs,9C1,83,27,23,23
Melissa,Black,9C1,34,42,43,21
Cynthia,Gonzales,9B1,19,41,25,100
Roger,Knight,9B1,99,47,47,25
```

Python cuenta con diferentes **funciones para la manipulación de archivos** (creación, lectura, escritura, eliminación, actualización).

Archivos

Apertura de un archivo

Para comenzar a utilizar un archivo se tiene que utilizar la función **open()**.

Función que permite abrir un archivo

Nombre del archivo

```
01. | archivo = open(archivo [, modo_de_acceso])
```

Nombre de la variable

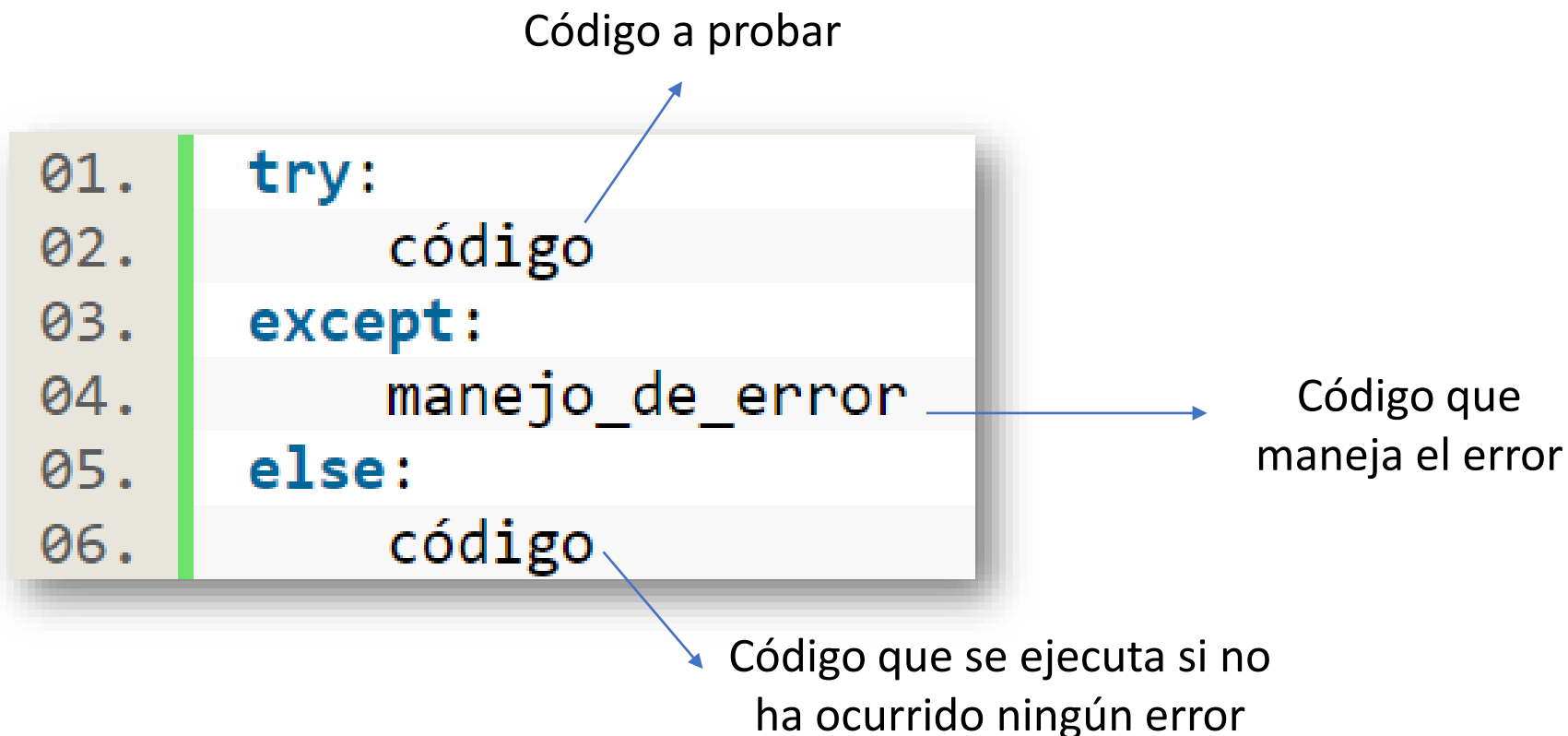
Modo de acceso al archivo

r	Lectura
r+	Lectura y escritura
w	Escritura
w+	Escritura y lectura

Archivos

Manejo de errores

Se utiliza el bloque **try..except** para manejar errores que ocurren durante la ejecución de un programa.



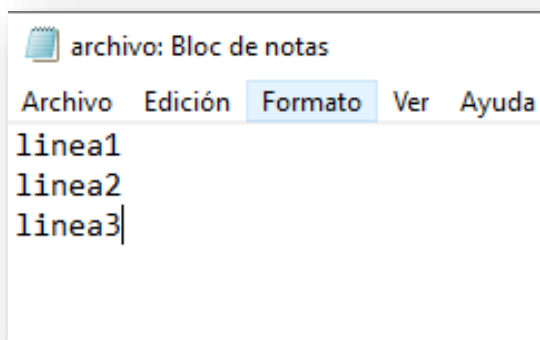
Archivos

Apertura de un archivo



Crear un archivo.

Crear un archivo con un editor de texto, escribir las siguientes líneas de código y guardar el archivo.



PyCharm.

Escribir las siguientes líneas de código:

```
try:
    archivo = open("ruta a su archivo de texto")
except OSError as err:
    print("Error: " + str(err))
else:
    archivo.close()
```

Colocar la ruta al archivo de texto.

Archivos

Atributos



Código:

Colocar dentro del *else*, antes de la línea de código *archivo.close()*

```
01. print(archivo.name) # Obtiene el nombre del archivo
02. print(archivo.mode) # Obtiene el modo con el que se abrió el archivo
03. print(archivo.closed) # Indica si el archivo está cerrado
```

Ejecución del código

```
C:\Users\yunue\Documents\archivo.txt
r
False
```

Cambiará dependiendo
de la ruta donde tengan
su archivo

Archivos

Leer contenido de un archivo línea por línea



Código

Colocar dentro del *else*, antes de la línea de código *archivo.close()*

```
01. for linea in archivo:  
02.     print(linea)
```

Ejecución del código

```
linea1  
  
linea2  
  
linea3
```

Archivos

Método read()



Código

Colocar dentro del *else*, antes de la línea de código *archivo.close()*. Comentar las líneas de código de la diapositiva anterior.

```
01. | print(archivo.read()) # Lee todo el contenido del archivo
```

Ejecución del código

```
linea1  
linea2  
linea3  
linea4
```

Archivos

Método read()

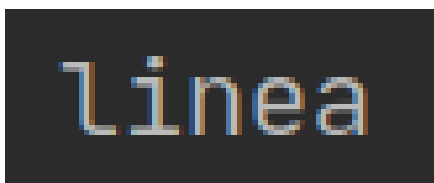


Código

Colocar dentro del *else*, antes de la línea de código *archivo.close()*. Comentar la línea de código de la diapositiva anterior

```
01. | #print(archivo.read()) # Lee todo el contenido del archivo
02. | print(archivo.read(5)) # Lee hasta el número de caracter indicado
```

Ejecución del código



Archivos

Método readline()



Código

Colocar dentro del *else*, antes de la línea de código *archivo.close()*. Comentar la línea de código de la diapositiva anterior

```
01. #print(archivo.read()) # Lee todo el contenido del archivo
02. #print(archivo.read(5)) # Lee hasta el número de caracter indicado
03. print(archivo.readline()) # Lee una línea del archivo
```

Ejecución del código

```
linea1
```

Archivos

Escritura de un archivo



Código

```
01. try:
02.     archivo2 = open("C:\\ruta_al_archivo\\archivo2.txt", "w+")
03.     archivo2.write("Hola") # Escribe en el archivo el texto indicado
04.     archivo2.write(" mundo!\n")
05.     archivo2.seek(0) # Cambia la posición al inicio del archivo
06.     print(archivo2.read())
07. except OSError as err:
08.     print("Error: " + str(err))
09. else:
10.     archivo2.close()
```

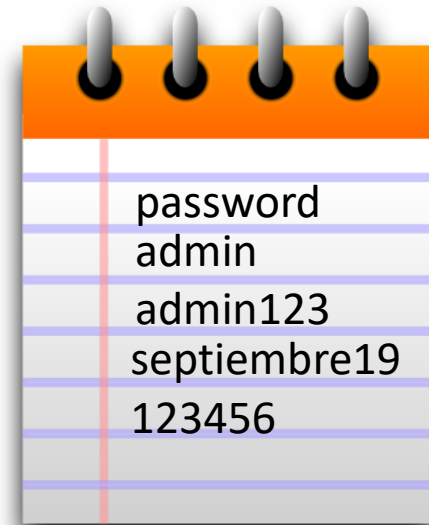
Ejecución del código

```
Hola mundo!
```

Ataques de diccionario

Ataque de diccionario

Ataque a hashes (utilizados para representar contraseñas) que **prueba una lista de posibles contraseñas** que se encuentran en un archivo llamado “diccionario”.



Hash

Cadena hexadecimal de tamaño fijo que representa una **huella digital o resumen de la información**.

“La junta será a
las 4:00 p.m.”



3ba5ba9d2ba08ccfacbea33c0e58fb7b

“password”



5f4dcc3b5aa765d61d8327deb882cf99

Hash

Funciones

Existen diversas funciones de resumen que permiten determinar el **valor hash** de un mensaje.

Función	Salida	Ejemplo
MD5	128 bits	5F4DCC3B5AA765D61D8327DEB882CF99

Hash

Funciones

Existen diversas funciones de resumen que permiten determinar el **valor hash** de un mensaje.

Función	Salida	Ejemplo
MD5	128 bits	5F4DCC3B5AA765D61D8327DEB882CF99
SHA-1	160 bits	5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8

Hash

Funciones

Existen diversas funciones de resumen que permiten determinar el **valor hash** de un mensaje.

Función	Salida	Ejemplo
MD5	128 bits	5F4DCC3B5AA765D61D8327DEB882CF99
SHA-1	160 bits	5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8
SHA-2	224, 256, 384, 512 bits	5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8

Hash

Funciones

Existen diversas funciones de resumen que permiten determinar el **valor hash** de un mensaje.

Función		Salida	Ejemplo
MD5		128 bits	5F4DCC3B5AA765D61D8327DEB882CF99
SHA-1		160 bits	5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8
SHA-2	224, 256, 384, 512 bits		5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D8
SHA-3	224, 256, 384, 512 bits		C0067D4AF4E87F00DBAC63B6156828237059172D1BBEAC67427345D6A9FDA484

Ataque de diccionario

Hashes a romper

5f4dcc3b5aa765d61d8327deb882cf99
e10adc3949ba59abbe56e057f20f883e

Diccionario de contraseñas

Qwerty123
password
123456
marzo2020
admin

2af9b1ba42dc5eb01743e6b3759b6e4b



Por cada contraseña en el diccionario se calcula su hash y se compara con los hashes a romper. Si son iguales, se ha identificado la contraseña.

Ataque de diccionario

Hashes a romper

5f4dcc3b5aa765d61d8327deb882cf99
e10adc3949ba59abbe56e057f20f883e

Diccionario de contraseñas

Qwerty123
password
123456
marzo2020
admin

2af9b1ba42dc5eb01743e6b3759b6e4b

5f4dcc3b5aa765d61d8327deb882cf99



5f4dcc3b5aa765d61d8327deb882cf99:password

Por cada contraseña en el diccionario se calcula su hash y se compara con los hashes a romper. Si son iguales, se ha identificado la contraseña.

Hash cracker

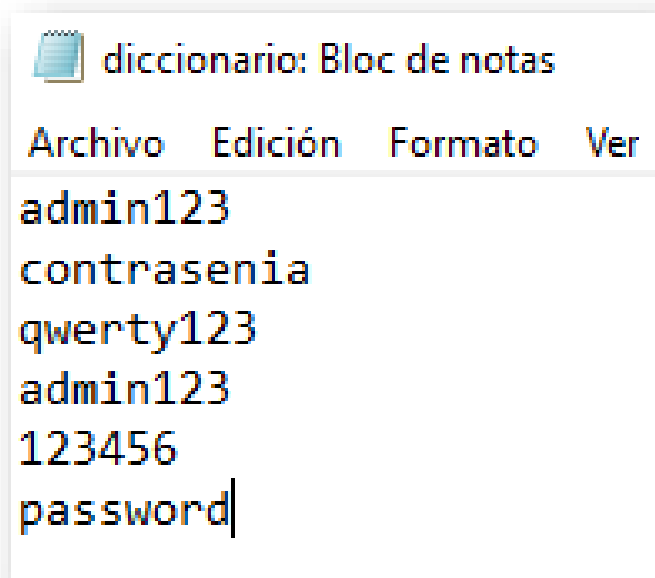
Diccionario

Primero debemos de crear el diccionario que vamos a utilizar en el script.



Crear el diccionario.

Abrir un editor de texto, escribir las siguientes contraseñas y guardar el archivo.



```
diccionario: Bloc de notas
Archivo  Edición  Formato  Ver
admin123
contrasenia
qwerty123
admin123
123456
password|
```

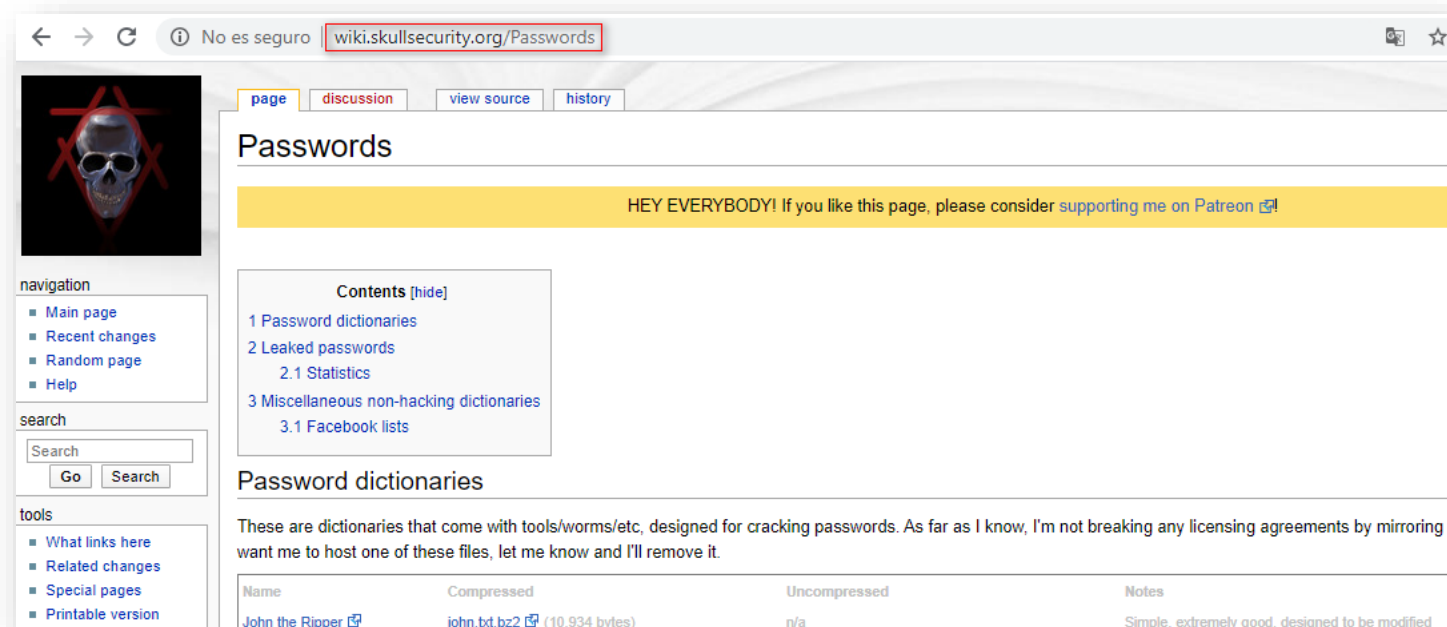

Script

```
01. import hashlib 1
02.
03. # Hash que se quiere crackear o romper
04. hash_a_romper = "5f4dcc3b5aa765d61d8327deb882cf99"
05.
06. # Ruta del diccionario a utilizar
07. ruta = "C:\\ruta_al_archivo\\diccionario.txt"
08.
09. try:
10.     # Abrir archivo del diccionario
11.     diccionario = open(ruta, "r") 2
12. except OSError as err:
13.     # Hubo un error a la hora de abrir el archivo
14.     print("Error: " + str(err))
15.     quit()
16. else:
17.     print("\n== Comenzando el crackeo ==")
18.     # Toma cada contraseña en el diccionario, calcula su hash y lo compara con
19.     for contrasenia in diccionario:
20.         print(" [+] Probando: " + contrasenia.strip())
21.         # Calcula el hash de la contraseña en el diccionario
22.         hash_c = hashlib.md5(contrasenia.encode('utf-8').strip()).hexdigest()
23.         # Compara si el hash calculado es el mismo que el se quiere romper
24.         if hash_c.strip() == hash_a_romper.strip():
25.             print("\nContraseña encontrada: " + contrasenia)
26.             break 5
27.
28.     # Cerrar el archivo
29.     diccionario.close()
```

- 1 *hashlib* es una librería de Python que permite utilizar las diferentes funciones hash que existen (MD5, SHA1, SHA2, etc.)
- 2 `open` es una función que permite leer el contenido de un archivo.
- 3 El ciclo `for` toma cada contraseña del diccionario para:
 - 4 Calcular hash de la contraseña obtenida del diccionario.
 - 5 Compararlo con el hash a romper

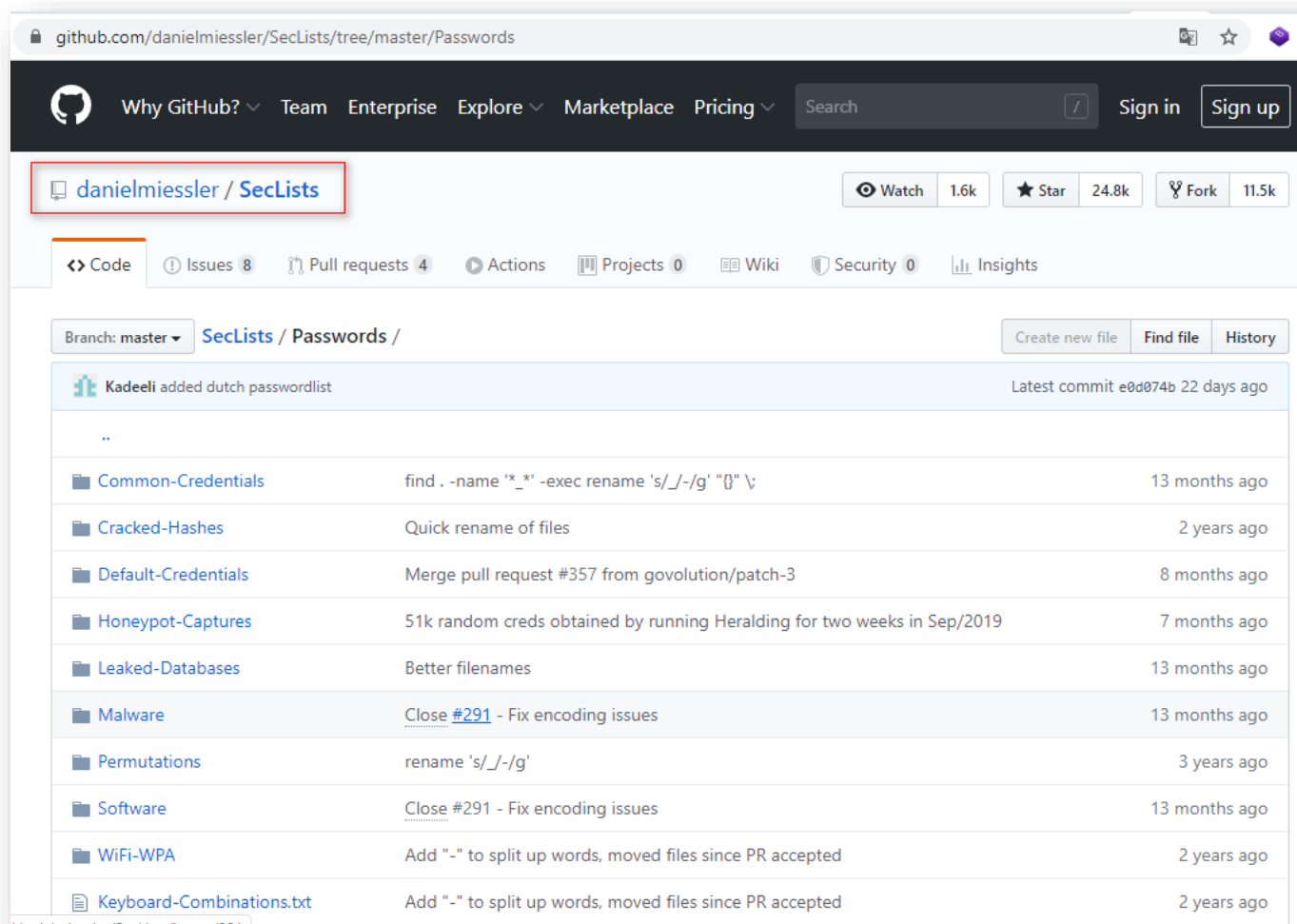
Diccionarios de contraseñas

Existen diccionarios de contraseñas que se pueden **descargar de internet**. Estos son alimentados con contraseñas comúnmente usadas u obtenidas de sitios que han sido vulnerados.



<https://wiki.skullsecurity.org/Passwords>

Diccionarios de contraseñas



The screenshot shows the GitHub web interface for the repository `danielmiessler/SecLists`, specifically the `tree/master/Passwords` view. The repository has 1.6k watches, 24.8k stars, and 11.5k forks. The `Code` tab is selected, showing a list of files and folders under the `Passwords` directory. The latest commit is by Kadeeli, adding a dutch passwordlist, 22 days ago.

File/Folder	Description	Time Ago
..		
Common-Credentials	<code>find . -name '*_*' -exec rename 's/_/-/g' {} \;</code>	13 months ago
Cracked-Hashes	Quick rename of files	2 years ago
Default-Credentials	Merge pull request #357 from govolution/patch-3	8 months ago
Honeypot-Captures	51k random creds obtained by running Heraldng for two weeks in Sep/2019	7 months ago
Leaked-Databases	Better filenames	13 months ago
Malware	Close #291 - Fix encoding issues	13 months ago
Permutations	<code>rename 's/_/-/g'</code>	3 years ago
Software	Close #291 - Fix encoding issues	13 months ago
WiFi-WPA	Add "-" to split up words, moved files since PR accepted	2 years ago
Keyboard-Combinations.txt	Add "-" to split up words, moved files since PR accepted	2 years ago

<https://github.com/danielmiessler/SecLists/tree/master/Passwords>

Próxima clase...

- **Capítulo 4:** Hands-On (Parte II)
- **Evaluación** (Capítulo 2)



NEXT >>
Mar, 25 Ago



CloudLamb

**¡Muchas gracias por su
atención!**