Report S6/L5

Esercizio del Giorno

Si ricordi che la configurazione dei servizi costituisce essa stessa una parte integrante dell'esercizio.

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:

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

Esercizio fase 2 – suggerimento:

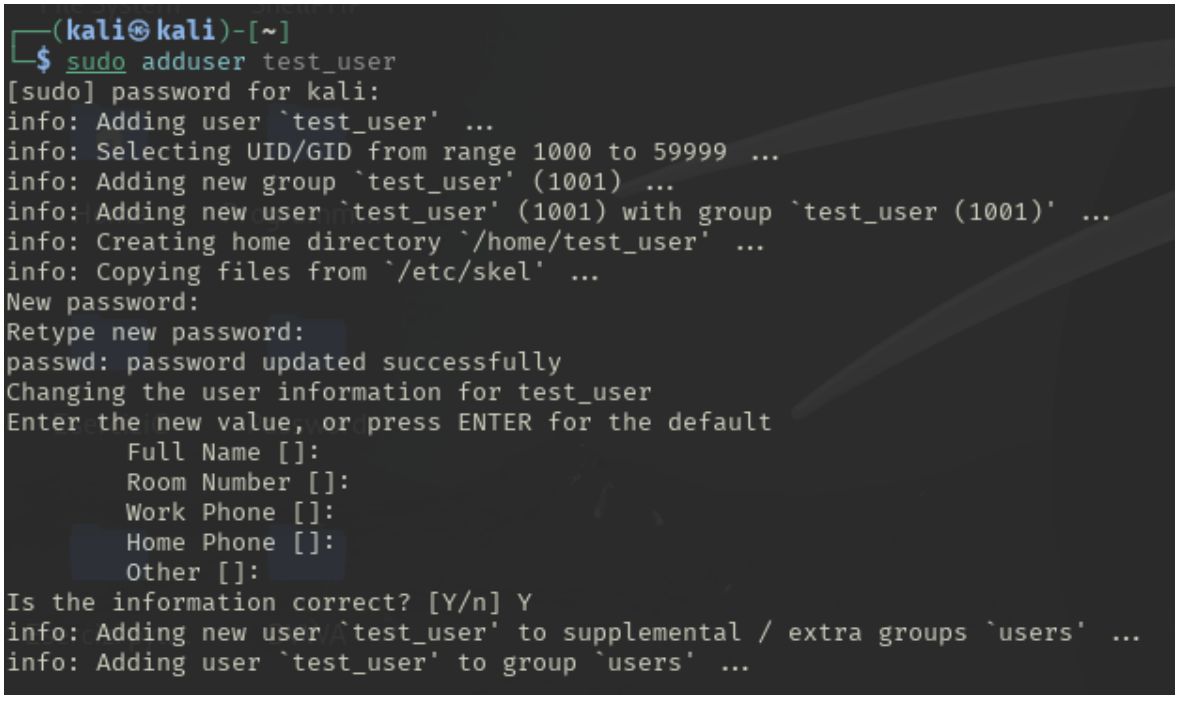
Per la seconda parte dell'esercizio, scegliete un servizio da configurare e poi provate a craccare l'autenticazione con Hydra.

- Se optate per il servizio ftp, potete semplicemente installarlo con il seguente comando:
  sudo apt-get install vsftpd

- E poi avviare il servizio con: **service vsftpd start**

Svolgimento

Per iniziare ho creato il nuovo utente su kali con il comando < **sudo adduser** > e l'ho chiamato **test_user** ho confermato inserendo la psw del mio kali e ho impostato **testpass** come password per in nuovo account, ho lasciato le altre informazioni come di default.



Ho attivato il servizio **ssh** con il comando < **sudo service ssh start** >.

Poi ho ispezionato il file sshd_config nella directory /etc/ssh/sshd_config.

```
  GNU nano 8.2                                 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
#HostbasedAuthentication no
```

```
  GNU nano 8.2                                 sshd_config
# 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 challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

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

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin prohibit-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
```

```
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem       sftp    /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
```

L’esercizio prevede di lasciare le configurazioni di default.

Per connettermi in SSH all'utente appena creato uso il comando <**ssh test_user@192.168.30.3**> dove:

-ssh → servizio

-test_user → utente creato in precedenza

-192.168.30.3 → IP kali

```
┌──(kali㉿kali)-[~]
└─$ ssh test_user@192.168.30.3
The authenticity of host '192.168.30.3 (192.168.30.3)' can't be established.
ED25519 key fingerprint is SHA256:o516ZakcJ5xEXJc5+cNr+F5Yh/7qf8AnZxXmH5LKtcg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.30.3' (ED25519) to the list of known hosts.
test_user@192.168.30.3's password:
Connection closed by 192.168.30.3 port 22
```

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

┌──(kali㉿kali)-[~]
└─$ ssh test_user@192.168.30.3
test_user@192.168.30.3's password:
Linux kali 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15) 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)-[~]
└─$
```

Mentre per tornare all'utente originario cambio semplicemente il nome dell'utente ovvero:

-ssh kali@192.168.30.3

```
┌──(test_user㉿kali)-[~]
└─$ ssh kali@192.168.30.3
The authenticity of host '192.168.30.3 (192.168.30.3)' can't be established.
ED25519 key fingerprint is SHA256:o516ZakcJ5xEXJc5+cNr+F5Yh/7qf8AnZxXmH5LKtcg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? Y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.30.3' (ED25519) to the list of known hosts.
kali@192.168.30.3's password:
Linux kali 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15) 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.
┌──(kali㉿kali)-[~]
└─$
```

Per craccare username e password usiamo il software Hydra con il seguente comando:

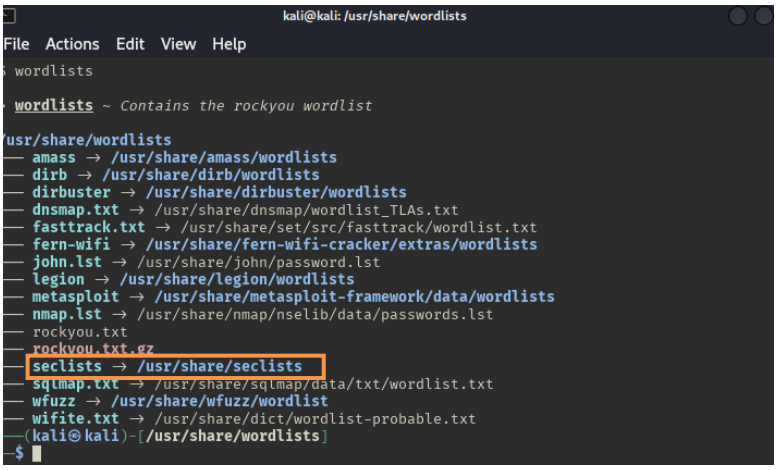
hydra -L username_list -P password_list IP_KALI –t 4 ssh

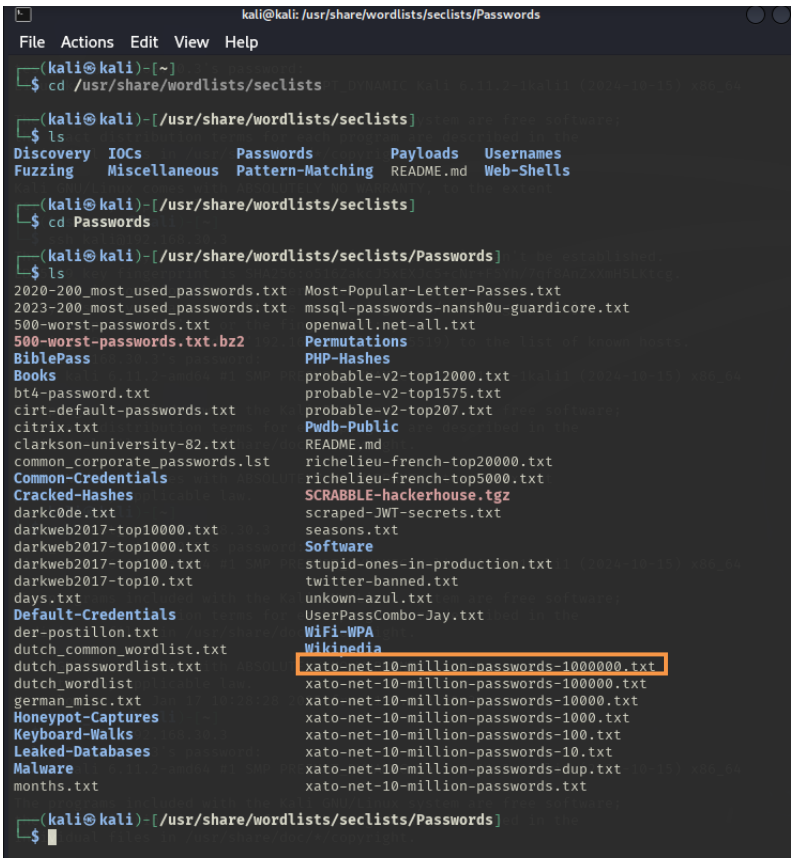
dove:

-L → per selezionare un file contenente la lista di possibili username

-P → per selezionare un file contenente la lista di possibili password

Le directory dei file le abbiamo trovate grazie al tool wordlist



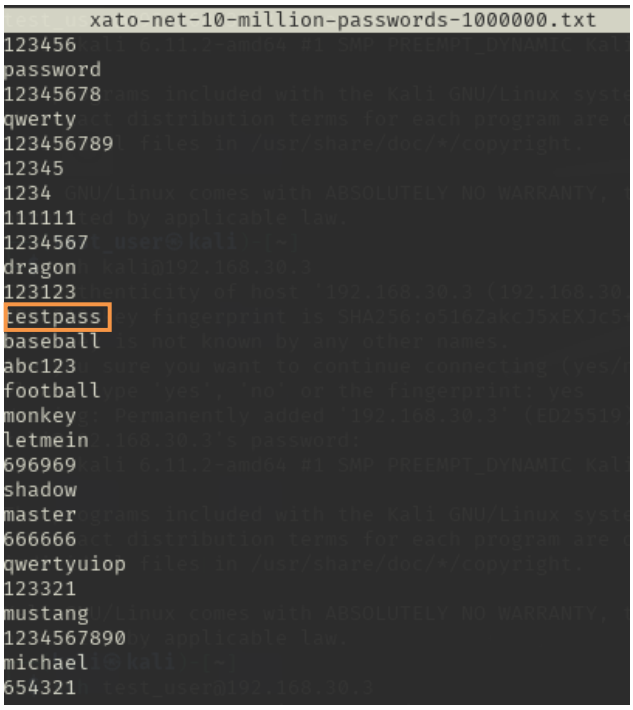


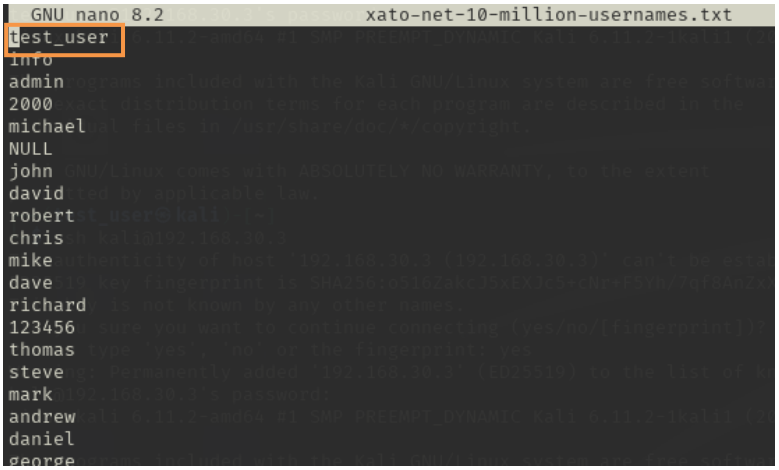
I file che utilizzerò sono i seguenti:

xato-net-10-million-passwords-1000000.txt

xato-net-10-million-usernames.txt

Al fine di ridurre le tempistiche ho inserito nelle prime posizioni delle liste il mio username e la mia password.

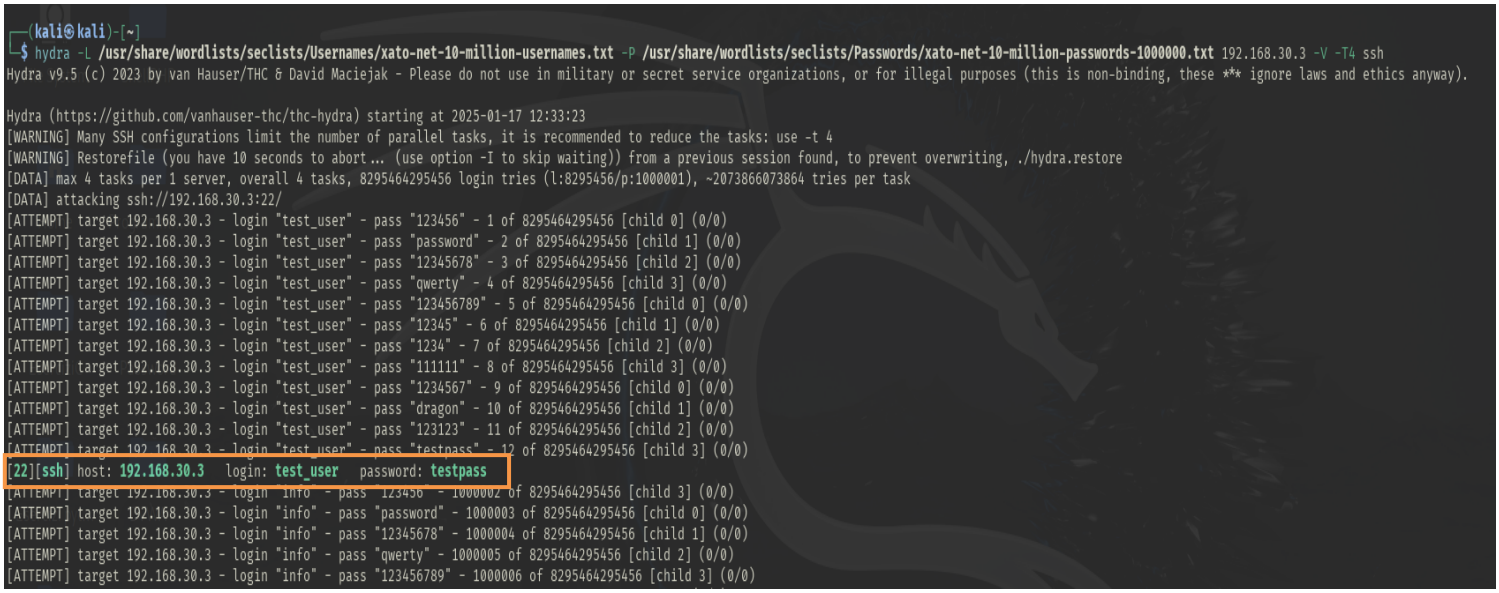




Il comando quindi diventa:

hydra -L /usr/share/wordlists/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.30.3 –t 4 ssh

Possiamo aggiungere lo **switch –V** per controllare il progresso del cracking, ecco il risultato:



Hydra ha trovato le credenziali nelle sue liste e me le ha evidenziate nella ricerca.

Nel risultato notiamo:

22 → porta

Ssh → protocollo

Host → il nostro IP

Login → username

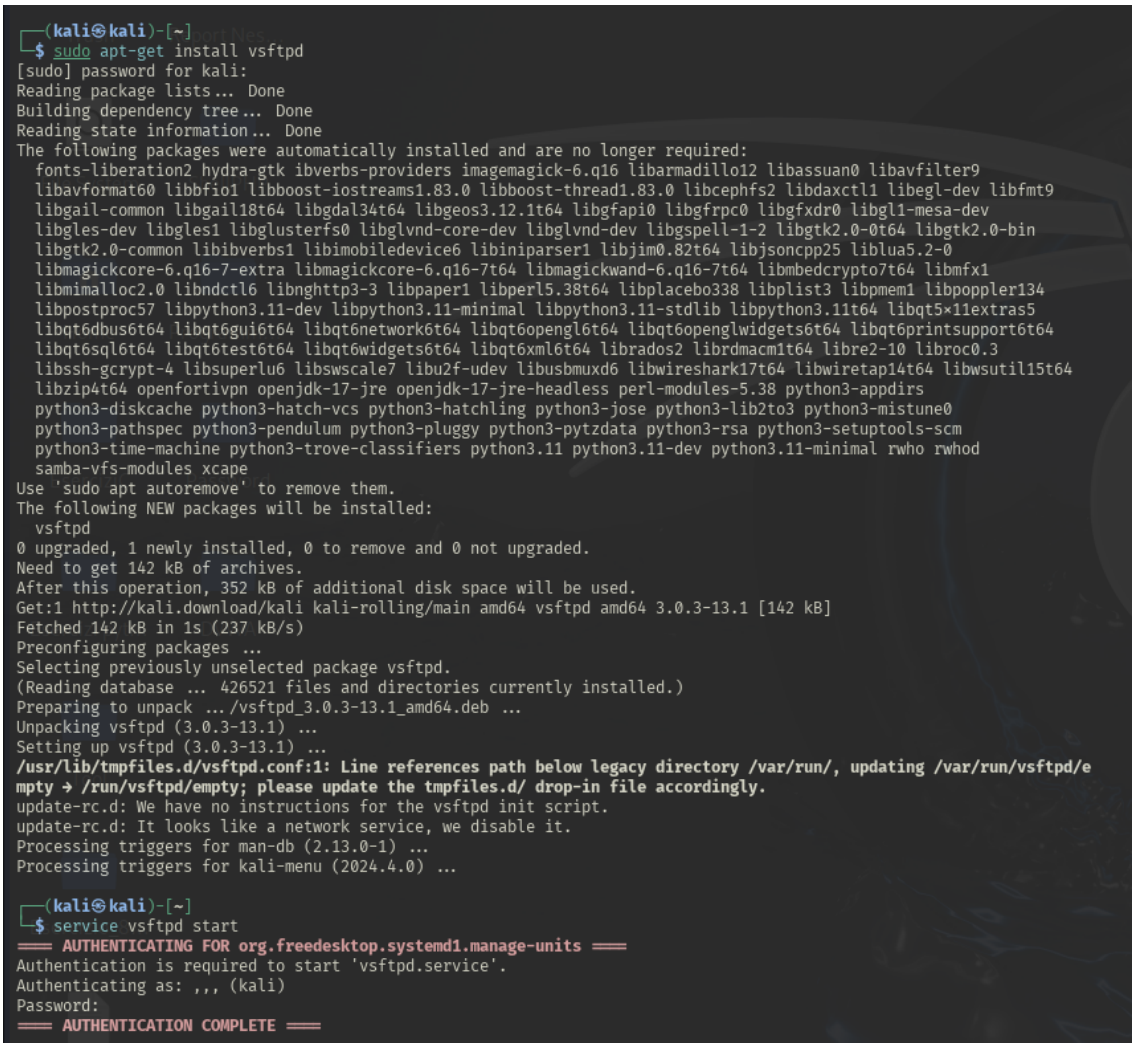
Password → la nostra password selezionata per il test

Terminata la configurazione e il cracking delle credenziali, inizio la seconda parte dell'esercizio.

Esercizio 2

Installiamo il servizio FTP con il comando: **sudo apt-get install vsftpd**

Avviamo il servizio con il comando: **service vsftpd start**



Mandiamo il comando:

- **hydra -L /usr/share/wordlists/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-1000000.txt 192.168.30.3 –V –t 4 ftp**

Anche in questo caso notiamo che **hydra** è stata in grado di rilevare le credenziali sfruttando il **protocollo ftp** sulla **porta 21**.