

Week 6 - Esercizio 4

Esercizio sul Password Cracking con Hydra

Procedimento

Vado innanzitutto a creare un nuovo user che chiamo **kali_ssh**

```
(kali㉿kali)-[~/Desktop]
└─$ sudo adduser kali_ssh
[sudo] password for kali:
Adding user `kali_ssh' ...
Adding new group `kali_ssh' (1002) ...
Adding new user `kali_ssh' (1001) with group `kali_ssh' ...
Creating home directory `/home/kali_ssh' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for kali_ssh
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
```

Ho impostato la password dell'utente come **testpass** come richiesto dalla consegna

Non avevo la possibilità di avviare il servizio ssh dal nuovo utente appena creato, quindi sono andato nella cartella **sudoers.d** e gli ho garantito la possibilità di eseguire il comando **sudo** attraverso il file di configurazione **kali-grant-root**, necessario ad avviare il servizio ssh

```
(kali㉿kali)-[/etc/sudoers.d]
```

Ho aggiunto una riga al file per fare in modo che l'utente abbia questa possibilità

```
File Actions Edit View Help
GNU nano 6.4
# Allow members of group kali-trusted to execute
# password prompt
%kali-trusted ALL=(ALL:ALL) NOPASSWD: ALL
kali_ssh ALL=(ALL:ALL) NOPASSWD: ALL
```

Questo non era necessario ai fini della risoluzione dell'esercizio, ma sono incappato in questo problema e ho deciso di trovarne la soluzione.

A questo punto avvio quindi il servizio ssh:

```
(kali㉿kali)-[/etc/sudoers.d]
$ sudo service ssh start
```

E lo vado anche a configurare tramite il file `/etc/ssh/sshd_config`:

```
GNU nano 6.4
# 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
```

A questo punto torno sull'user principale **kali** e vado a testare la connessione ssh:

```
(kali㉿kali)-[/etc/sudoers.d]
$ ssh kali_ssh@10.0.2.15
```

E inserisco la password dell'utente per poter accedere

```
warning: Permanently added '10.0.2.15' (ED25519) to the list of known hosts.
kali_ssh@10.0.2.15's password:
Linux kali 5.18.0-kali5-amd64 #1 SMP PREEMPT_DYNAMIC Debian 5.18.5-1kali6 (2022-07-07) 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_ssh㉿kali)-[~]
$
```

Inserendo correttamente la password ho l'accesso al prompt dei comandi dell'utente **kali_ssh**

Ora che abbiamo configurato tutto correttamente, andiamo ad effettuare il cracking dell'accesso ssh con **Hydra**:

- **CRACKING SINGOLO:**

```
(kali㉿kali)-[/home/kali_ssh]
$ hydra -l kali_ssh -p testpass 10.0.2.15 -t 4 ssh
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or
and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-12-01 09:32:19
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://10.0.2.15:22/
[22][ssh] host: 10.0.2.15 login: kali_ssh password: testpass
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-12-01 09:32:19
```

- **CRACKING con WORDLISTs:**

Scarico la repository **seclists** che contiene le liste di password e nomi utente che ci interessano per craccare l'accesso del nostro utente su Kali

```
(kali㉿kali)-[~/Downloads]
$ git clone https://github.com/danielmiessler/SecLists
Cloning into 'SecLists'...
remote: Enumerating objects: 11968, done.
remote: Counting objects: 100% (82/82), done.
```

Lancio il comando e attendo che hydra trovi la corrispondenza:

```
(kali㉿kali)-[~]
$ hydra -V -L SecLists/Username/xato-net-10-million-usernames.txt -P SecLists/Passwords/xato-net-10-million-passwords-1000000.txt 10.0.2.15 -t 4 ssh
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations or for illegal purposes (this is
```

Notare che per fare prima ho inserito un file il nome utente **kali_ssh** per fare prima nella ricerca della combinazione corretta

ESERCIZIO: Parte 2

Per la parte 2 dell'esercizio andiamo a eseguir la stessa cosa ma con un servizio differente da **ssh**

Nel nostro caso quello che andiamo a utilizzare sarà il servizio **ftp**

```
(kali@kali)-[~]  
$ sudo apt install vsftpd
```

E lancio il servizio:

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

Dopo di che rilancio hydra con il seguente comando:

```
(kali@kali)-[~]  
$ hydra -V -L SecLists/Usernames/xato-net-10-million-usernames.txt -P SecLists/Passwords/xato-net-10-million-passwords-1000000.txt 10.0.2.15 ftp
```

Notiamo che in questo caso non abbiamo inviato usato lo switch **-T 4**, come hydra suggerisce nel caso dell'ssh