Hydra

Hydra is a fast network login cracker that supports numerous attack protocols. It is a versatile tool that can brute-force a wide range of services, including web applications, remote login services like SSH and FTP, and even databases.

Hydra's popularity stems from its:

- `Speed and Efficiency`: Hydra utilizes parallel connections to perform multiple login attempts simultaneously, significantly speeding up the cracking process.
- `Flexibility`: Hydra supports many protocols and services, making it adaptable to various attack scenarios.
- `Ease of Use`: Hydra is relatively easy to use despite its power, with a straightforward command-line interface and clear syntax.

Installation

Hydra often comes pre-installed on popular penetration testing distributions. You can verify its presence by running:

Hydra

```
MisaelMacias@htb[/htb]$ hydra -h
```

If Hydra is not installed or you are using a different Linux distribution, you can install it from the package repository:

Hydra

```
MisaelMacias@htb[/htb]$ sudo apt-get -y update
MisaelMacias@htb[/htb]$ sudo apt-get -y install hydra
```

Basic Usage

Hydra's basic syntax is:

Hydra

```
MisaelMacias@htb[/htb]$ hydra [login_options] [password_options] [attack_options] [service_op
```

Parameter	Explanation	Usage Example
`-l LOGIN` or `-L FILE`	Login options: Specify either a single username (`-l`) or a file containing a list of usernames (`-L`).	`hydra -l admin ...` or `hydra -L usernames.txt ...`
`-p PASS` or `-P FILE`	Password options: Provide either a single password (`-p`) or a file containing a list of passwords (`-P`).	`hydra -p password123 ...` or `hydra -P passwords.txt ...`
`-t TASKS`	Tasks: Define the number of parallel tasks (threads) to run, potentially speeding up the attack.	`hydra -t 4 ...`
`-f`	Fast mode: Stop the attack after the first successful login is found.	`hydra -f ...`
`-s PORT`	Port: Specify a non-default port for the target service.	`hydra -s 2222 ...`
`-v` or `-V`	Verbose output: Display detailed information about the attack's progress, including attempts and results.	`hydra -v ...` or `hydra -V ...` (for even more verbosity)
`service://server`	Target: Specify the service (e.g., `ssh`, `http`, `ftp`) and the target server's address or hostname.	`hydra ssh://192.168.1.100`
`/OPT`	Service-specific options: Provide	`hydra http-get://example.com/login.php -m`

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

		any additional options required by the target service.	`"POST:user=^USER^&pass=^PASS^"` (for HTTP form-based authentication)

Hydra Services

Hydra services essentially define the specific protocols or services that Hydra can target. They enable Hydra to interact with different authentication mechanisms used by various systems, applications, and network services. Each module is designed to understand a particular protocol's communication patterns and authentication requirements, allowing Hydra to send appropriate login requests and interpret the responses. Below is a table of commonly used services:

Hydra Service	Service/Protocol	Description	Example Command
ftp	File Transfer Protocol (FTP)	Used to brute-force login credentials for FTP services, commonly used to transfer files over a network.	`hydra -l admin -P /path/to/password_list.txt ftp://192.168.1.100`
ssh	Secure Shell (SSH)	Targets SSH services to brute-force credentials, commonly used for secure remote login to systems.	`hydra -l root -P /path/to/password_list.txt ssh://192.168.1.100`
http-get/post	HTTP Web Services	Used to brute-force login credentials for HTTP web login forms using either GET or POST requests.	`hydra -l admin -P /path/to/password_list.txt http-post-form "/login.php:user=^USER^&pass=^PASS^:F=incorrect"`
smtp	Simple Mail Transfer Protocol	Attacks email servers by brute-forcing login credentials for SMTP, commonly used to send emails.	`hydra -l admin -P /path/to/password_list.txt smtp://mail.server.com`
pop3	Post Office Protocol (POP3)	Targets email retrieval services to brute-force credentials for POP3 login.	`hydra -l user@example.com -P /path/to/password_list.txt pop3://mail.server.com`
imap	Internet Message Access Protocol	Used to brute-force credentials for IMAP services, which allow users to access their email remotely.	`hydra -l user@example.com -P /path/to/password_list.txt imap://mail.server.com`
mysql	MySQL Database	Attempts to brute-force login credentials for MySQL databases.	`hydra -l root -P /path/to/password_list.txt mysql://192.168.1.100`
mssql	Microsoft SQL Server	Targets Microsoft SQL servers to brute-force database login credentials.	`hydra -l sa -P /path/to/password_list.txt mssql://192.168.1.100`
vnc	Virtual Network Computing (VNC)	Brute-forces VNC services, used for remote desktop access.	`hydra -P /path/to/password_list.txt vnc://192.168.1.100`
rdp	Remote Desktop Protocol (RDP)	Targets Microsoft RDP services for remote login brute-forcing.	`hydra -l admin -P /path/to/password_list.txt rdp://192.168.1.100`

Brute-Forcing HTTP Authentication

Imagine you're tasked with testing the security of a website using basic HTTP authentication at `www.example.com`. You have a list of potential usernames stored in `usernames.txt` and corresponding passwords in `passwords.txt`. To launch a brute-force attack against this HTTP service, use the following Hydra command:

Hydra

```
MisaelMacias@htb[/htb]$ hydra -L usernames.txt -P passwords.txt www.example.com http-get
```

This command instructs Hydra to:

- Use the list of usernames from the `usernames.txt` file.
- Use the list of passwords from the `passwords.txt` file.
- Target the website `www.example.com`.
- Employ the `http-get` module to test the HTTP authentication.

Hydra will systematically try each username-password combination against the target website to discover a valid login.

Targeting Multiple SSH Servers

Consider a situation where you have identified several servers that may be vulnerable to SSH brute-force attacks. You compile their IP addresses into a file named `targets.txt` and know that these servers might use the default username "root" and password "toor." To efficiently test all these servers simultaneously, use the following Hydra command:

Hydra

```
MisaelMacias@htb[/htb]$ hydra -l root -p toor -M targets.txt ssh
```

This command instructs Hydra to:

- Use the username "root".
- Use the password "toor".
- Target all IP addresses listed in the `targets.txt` file.
- Employ the `ssh` module for the attack.

Hydra will execute parallel brute-force attempts on each server, significantly speeding up the process.

Testing FTP Credentials on a Non-Standard Port

Imagine you need to assess the security of an FTP server hosted at `ftp.example.com`, which operates on a non-standard port `2121`. You have lists of potential usernames and passwords stored in `usernames.txt` and `passwords.txt`, respectively. To test these credentials against the FTP service, use the following Hydra command:

Hydra

```
MisaelMacias@htb[/htb]$ hydra -L usernames.txt -P passwords.txt -s 2121 -V ftp.example.com ft
```

This command instructs Hydra to:

- Use the list of usernames from the `usernames.txt` file.
- Use the list of passwords from the `passwords.txt` file.
- Target the FTP service on `ftp.example.com` via port `2121`.
- Use the `ftp` module and provide verbose output (`-V`) for detailed monitoring.

Hydra will attempt to match each username-password combination against the FTP server on the specified port.

Brute-Forcing a Web Login Form

Suppose you are tasked with brute-forcing a login form on a web application at `www.example.com`. You know the username is "admin," and the form parameters for the login are `user=^USER^&pass=^PASS^`. To perform this attack, use the following Hydra command:

Hydra

```
MisaelMacias@htb[/htb]$ hydra -l admin -P passwords.txt www.example.com http-post-form "/logi
```

This command instructs Hydra to:

- Use the username "admin".
- Use the list of passwords from the `passwords.txt` file.
- Target the login form at `/login` on `www.example.com`.
- Employ the `http-post-form` module with the specified form parameters.

- Look for a successful login indicated by the HTTP status code `302`.

Hydra will systematically attempt each password for the "admin" account, checking for the specified success condition.

Advanced RDP Brute-Forcing

Now, imagine you're testing a Remote Desktop Protocol (RDP) service on a server with IP `192.168.1.100`. You suspect the username is "administrator," and that the password consists of 6 to 8 characters, including lowercase letters, uppercase letters, and numbers. To carry out this precise attack, use the following Hydra command:

Hydra

```
MisaelMacias@htb[/htb]$ hydra -l administrator -x 6:8:abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLM
```

This command instructs Hydra to:

- Use the username "administrator".
- Generate and test passwords ranging from 6 to 8 characters, using the specified character set.
- Target the RDP service on `192.168.1.100`.
- Employ the `rdp` module for the attack.

Hydra will generate and test all possible password combinations within the specified parameters, attempting to break into the RDP service.