Infinite Logins

HACKING TUTORIAL, PENTESTING

How to Brute Force Websites & Online Forms Using Hydra

Posted on February 22, 2020July 29, 2021 by Harley in Hacking Tutorial, Pentesting

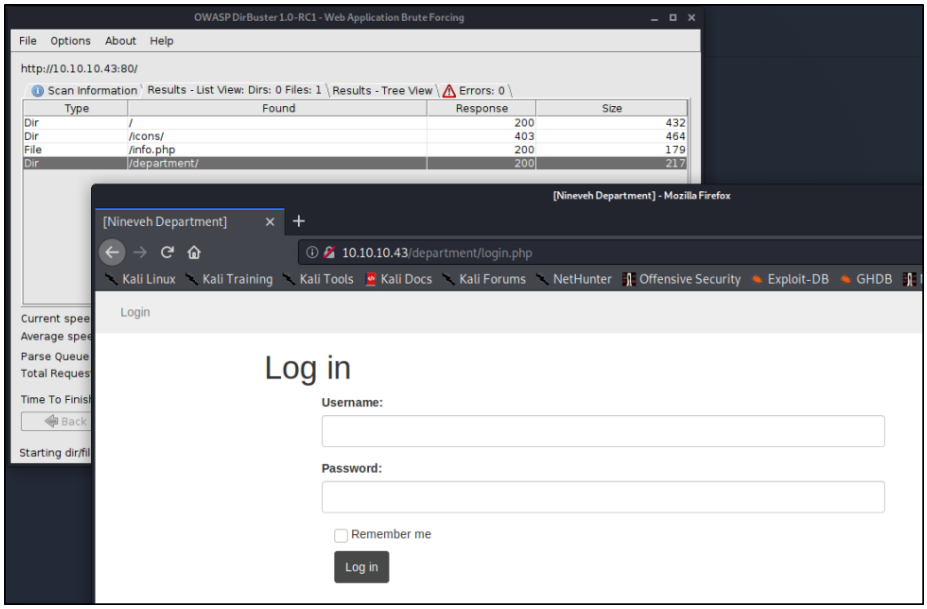
While working through NINEVAH on HackTheBack (Write-Up on this coming in a future post), I came across a couple web forms that I needed to break into. In my opinion, using the Intruder feature within BurpSuite is an easier way to run brute-force attacks, but the effectiveness of the tool is greatly reduced when using the free community version. Instead of dealing with slow brute-force attempts, I decided to give Hydra a try.

What we're breaking into

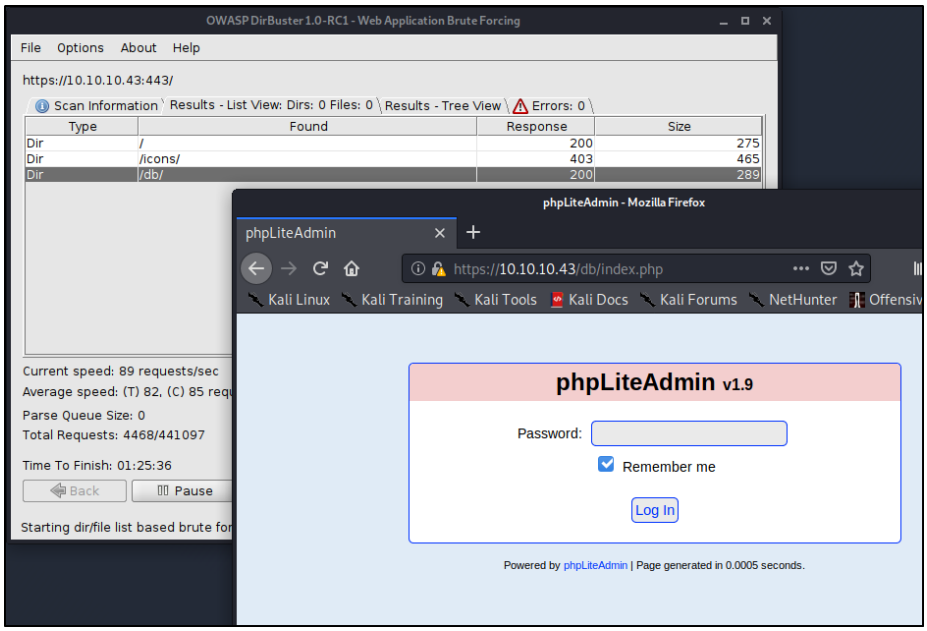
If you're unfamiliar with https://hackthebox.eu, I highly recommend checking them out. Click here to check out my Hack-TheBox related content.

NINEVAH sits on HackTheBox servers at IP address 10.1.10.43. I found a couple login pages at the following URLs. These are the addresses we're going to attempt to break into.

1st Address: http://10.10.10.43/department/login.php



2nd Address: https://10.10.10.43/db/index.php



Using Hydra to Brute-Force Our First Login Page

Hydra is a fairly straight forward tool to use, but we have to first understand what it needs to work correctly. We'll need to provide the following in order to break in:

- Login or Wordlist for Usernames
- Password or Wordlist for Passwords
- IP address or Hostname
- HTTP Method (POST/GET)
- Directory/Path to the Login Page
- Request Body for Username/Password
- A Way to Identify Failed Attempts

Let's start piecing together all the necessary flags before finalizing our command.

Specifying Username

In our particular case, we know that the username *Admin* exists, which will be my target currently. This means we'll want to use the `-l` flag for Login.

```
-l admin
```

Note: If you don't know the username, you could leverage `-L` *to provide a wordlist and attempt to enumerate usernames. This will only be effective if the website provides a way for you to determine correct usernames, such as saying "Incorrect Username" or "Incorrect Password", rather than a vague message like "Invalid Credentials".*

Specifying Password

We don't know the password, so we'll want to use a wordlist in order to perform a Dictionary Attack. Let's try using the common rockyou.txt list (by specifying a capital `-P`) available on Kali in the /usr/share/wordlists/ directory.

```
-P /usr/share/wordlists/rockyou.txt
```

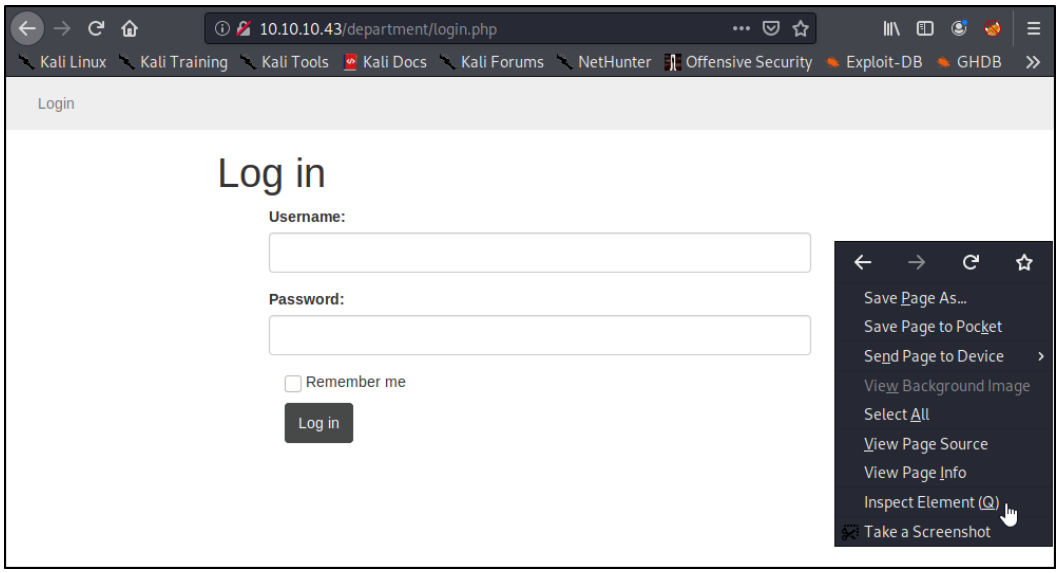
IP Address to Attack

This one is easy!

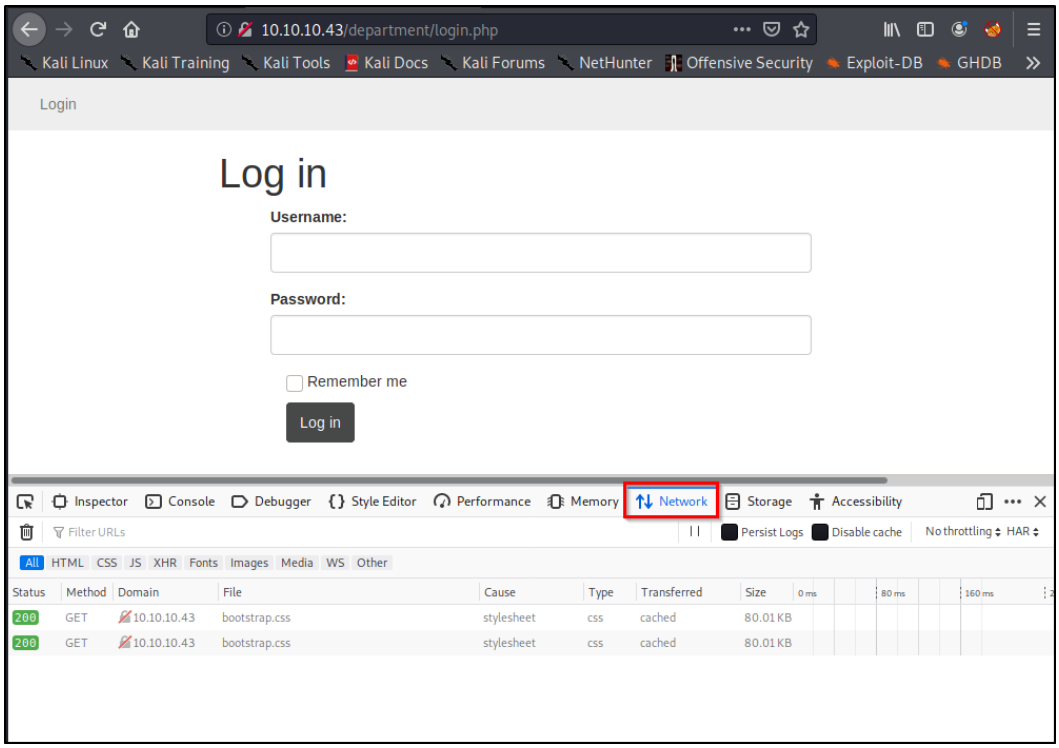
```
10.10.10.43
```

Specifying Method

This is where we need to start pulling details about the webpage. Let's head back into our browser, right-click, and **Inspect Element**.

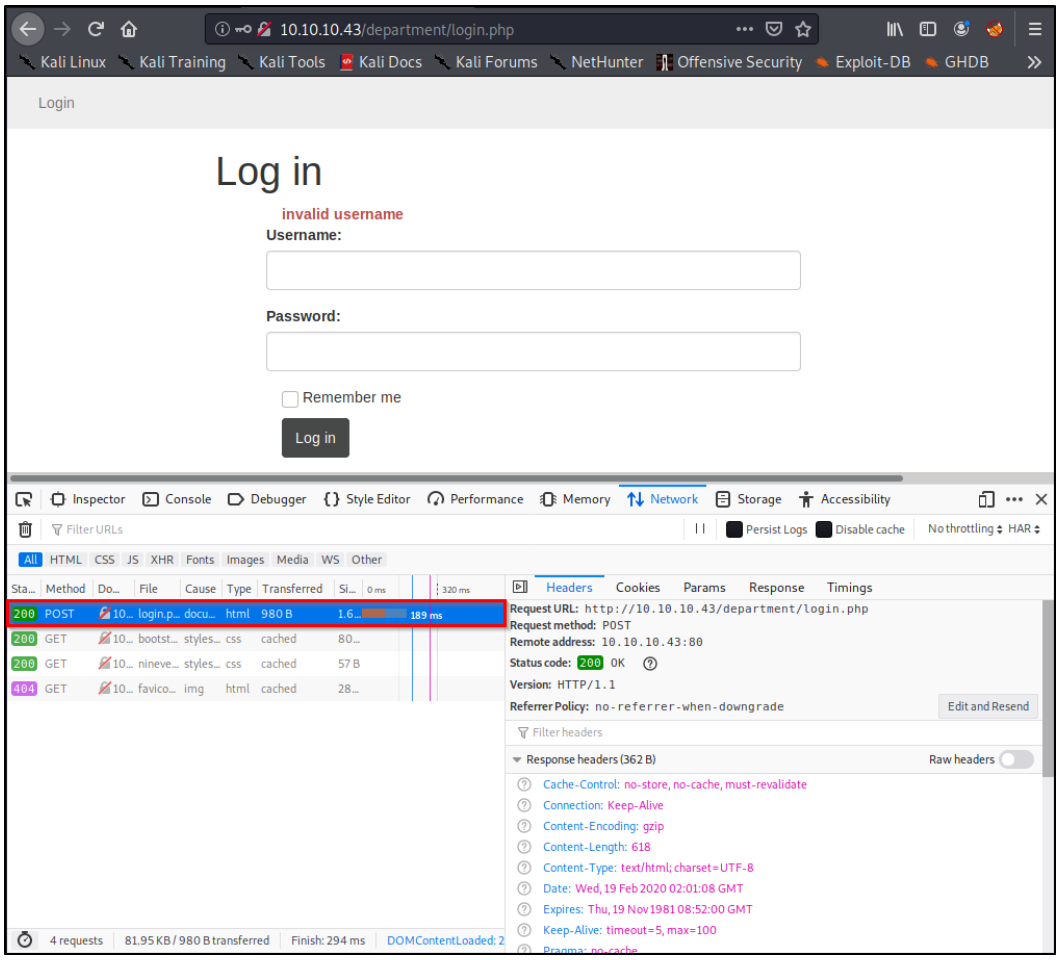


A window should pop-up on the bottom of the page. Go ahead and select the **Network** tab.



Right away, we see a couple GET methods listed here, but let's see what happens if we attempt a login. Go ahead and type in a random username/password, and click **Log In**.

Of course our login attempt will fail, but we're able to see that this website is using a **POST** method to log-in by looking at the requests.



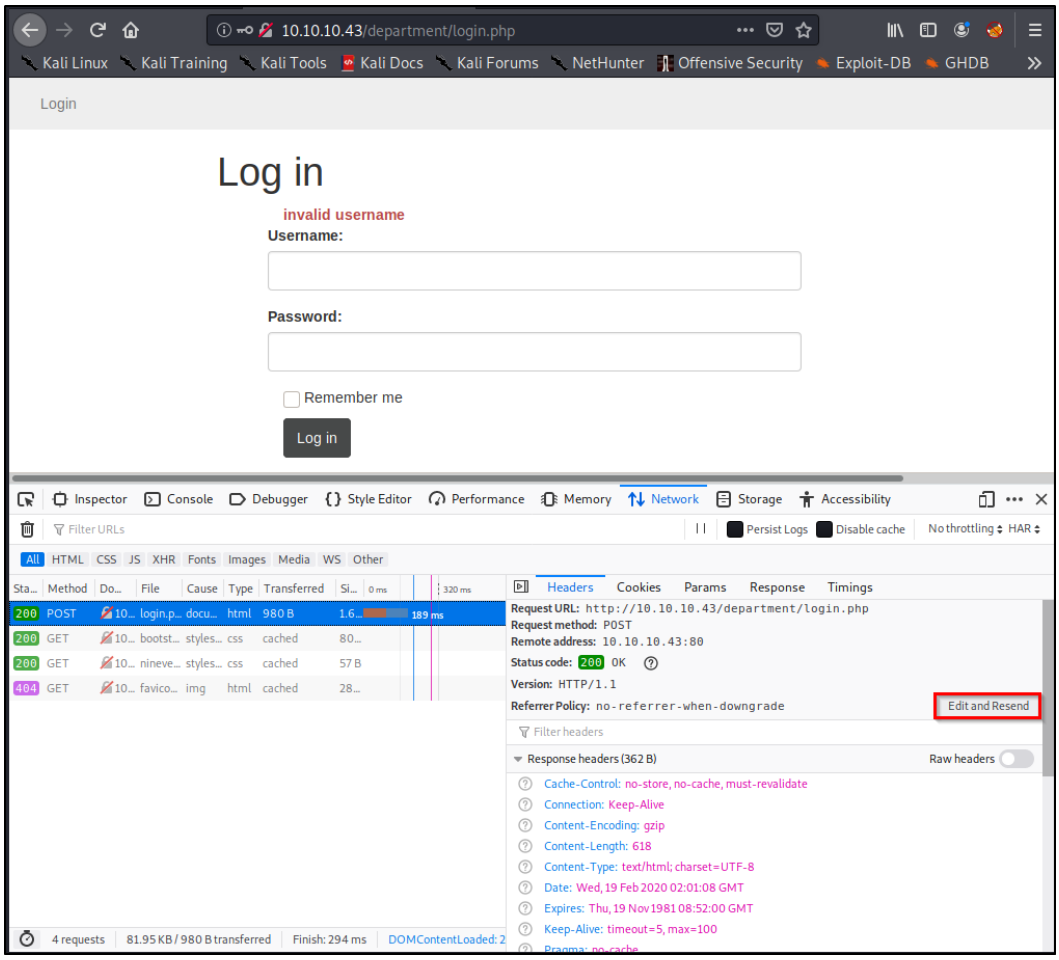
Easy enough, now we know what method to specify in our command!
`http-post-form`
Note: You'll need to enter https if you're attacking a site on port 443.

Specifying the Path to Attack

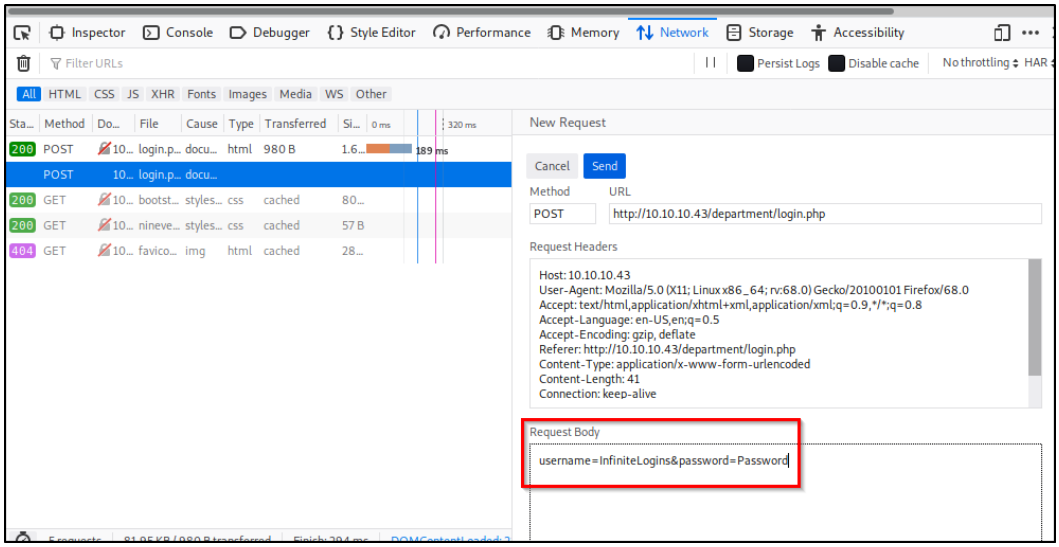
So far, we've only told the tool to attack the IP address of the target, but we haven't specified where the login page lives. Let's prepare that now.
`/department/login.php`

Finding & Specifying Location of Username/Password Form(s)

This is the hardest part, but it's actually surprisingly simple. Let's head back over to our browser window. We should still have the Inspect Element window open on the Network Tab. With our Post request still selected, let's click **Edit and Resend**.



Now we see a section called **Request Body** that contains the username and password you entered earlier! We'll want to grab this entire request for Hydra to use.



In my case, the unmodified request looks like this:
`username=InfiniteLogins&password=Password`

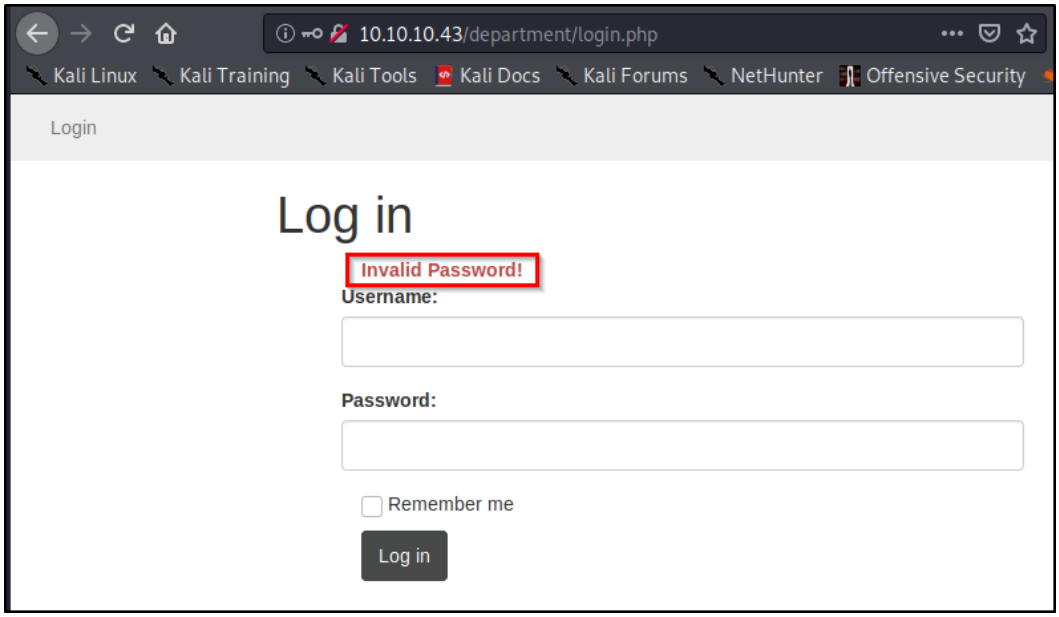
Because we know the username we're after is "admin", I'm going to hardcode that into the request. I'll also replace the "Password" I entered with ^PASS^. This will tell Hydra to enter the words from our list in this position of the request. My modified request that I'll place into my Hydra command looks like this:
`username=admin&password=^PASS^`

Note: If we desired, we could also brute-force usernames by specifying ^USER^ instead of admin.

Identifying & Specifying Failed Attempts

Finally, we just need a way to let Hydra know whether or not we successfully logged-in. Since we can't see what the page looks like upon a successful login, we'll need to specify what the page looks like on a failed login.

Let's head back to our browser and attempt to login using the username of *admin* and password of *password*.



As we saw before, we're presented with text that reads "Invalid Password!" Let's copy this, and paste it into our command:

```
Invalid Password!
```

Piecing the Command Together

Let's take all of the components mentioned above, but place them into a single command. Here's the syntax that we're going to need.

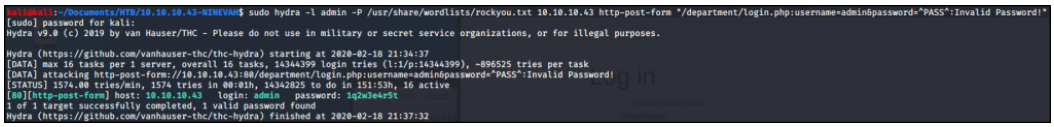
```
sudo hydra <Username/List> <Password/List> <IP> <Method> "<Path>:<RequestBody>:
<IncorrectVerbiage>"
```

After filling in the placeholders, here's our actual command!

```
sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.43 http-post-
form "/department/login.php:username=admin&password=^PASS^:Invalid Password!"
```

Note: I ran into issues later on when trying to execute this copied command out of this WordPress site. You may need to delete and re-enter your quotation marks within the terminal window before the command will work properly for you.

After a few minutes, we uncover the password to sign in!
`admin:1q2w3e4r5t`



Using Hydra to Brute-Force Our Second Login Page

Go through the exact same steps as above, and you should end up with a command that looks like this.
`sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.43 https-post-form "/db/index.php:password=^PASS^&remember=yes&login=Log+In&proc_login=true:Incorrect password"`

So what's different between this command and the one we ran earlier? Let's make note of the things that changed.

- Method was switched to `https-post-form`
- Path was updated to `/db/index.php`
- Request Body is completely different, but we still hard-code `admin` and replace the password with `^PASS^`
- Finally, the text returned for a failed attempt reads `Incorrect password`

After running the command, we uncover the password after just a couple minutes.
`admin:password123`

Let me know if you found this at all helpful, or if something didn't quite work for you!

BRUTE-FORCE **HACKING TUTORIAL** **HACKTHEBOX** **HYDRA** **NINEVAH** **THC-HYDRA**

4

4 thoughts on "How to Brute Force Websites & Online Forms Using Hydra"

1. Pingback: Hacking Methodology Cheatsheet | Infinite Logins

2. **Torch**
says:
November 3, 2020 at 11:37 pm
Thank you so much, the trick of clicking 'Edit and resend' to grab the request body was just the tip I needed!

Reply

3. **Yug**
says:
July 13, 2021 at 5:46 am
Hello, I appreciate you this article. I am seeking far more content similar to this. Kindly continue to keep updating

Reply

4. **jo**
says:
July 18, 2021 at 6:58 pm
grate

Reply