**HTB: SQLi Login Bypass**

**Overview:** A short, clear walkthrough of how I discovered and responsibly tested a comment-style SQL injection to bypass a login form in a lab environment. *This is written in a human tone and is redacted for public sharing—replace `TARGET\_IP` and `FLAG{REDACTED}` only with your own lab values.*

What I did (plain language):

**1) Find the web server**

I ran a quick service scan to see what was listening on the host:  
  
nmap -sV -p 80 TARGET\_IP  
A screen shot of a computer

AI-generated content may be incorrect.  
This confirmed port 80 was open and running an HTTP service.

**2) Enumerate web content**

I used Gobuster to look for common files and directories:  
  
gobuster dir -u http://TARGET\_IP -w /usr/share/dirb/wordlists/common.txt

A computer screen shot of a computer

AI-generated content may be incorrect.  
  
This showed /index.php returned HTTP 200, so I opened it in a browser.

**3) Observe the login form**

Visiting http://TARGET\_IP/index.php displayed a login page. I inspected the form to confirm standard username/password fields (and checked for CSRF tokens or hidden fields before automating).

A screenshot of a computer

AI-generated content may be incorrect.

**4) Try a safe comment-style SQLi bypass (lab only)**

A quick, low-impact test is to use a comment to neutralize the password check. In the username field I tried payloads such as:  
  
admin' --   
admin' #  
admin' /\*  
A screenshot of a computer

AI-generated content may be incorrect.  
In this lab, `admin' #` worked (MySQL treats `#` as a comment). Submitting that as username (password anything) caused the app to accept me as admin and showed a page containing the flag.

**5) Why that works (short explanation)**

If the app composes SQL like:  
  
SELECT \* FROM users WHERE username = '<INPUT>' AND password = '<PASS>';  
  
Then `admin' #` becomes:  
  
SELECT \* FROM users WHERE username = 'admin' # ' AND password = '...';  
  
Everything after `#` is ignored, so the password check is commented out and login succeeds.