

Our target machine here is 10.129.223.159 – as always, let's start with our basic nmap scan:

```
[us-starting-point-2-dhcp]~[10.10.14.203]~[emrom8@htb-wwcaodu5ip]~[~]
[*]$ nmap -p- -sS -sV 10.129.223.159
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-06 10:46 CDT
Nmap scan report for 10.129.223.159
Host is up (0.010s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.38 ((Debian))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.57 seconds
```

Let's see what exploits are potentially available for the service on port 80:

```
[us-starting-point-2-dhcp]~[10.10.14.203]~[emrom8@htb-wwcaodu5ip]~[~]
[*]$ searchsploit httpd 2.4.38
-----
Exploit Title | Path
-----
OpenBSD HTTPd < 6.0 - Memory Exhaustion Denial of Service | openbsd/dos/41278.txt
Shellcodes: No Results
[us-starting-point-2-dhcp]~[10.10.14.203]~[emrom8@htb-wwcaodu5ip]~[~]
[*]$ searchsploit httpd 2.4
-----
Exploit Title | Path
-----
Apache 2.4.23 mod_http2 - Denial of Service | linux/dos/40909.py
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE) | multiple/webapps/50383.sh
Omnicon OmniHTTPd 1.1/2.4 Pro - Remote Buffer Overflow | windows/remote/19566.c
OmniHTTPd 1.1/2.0.x/2.4 - 'test.php' Sample Application Cross-Site Scripting | windows/remote/21753.txt
OmniHTTPd 1.1/2.0.x/2.4 - Sample Application URL Encoded Newline HTML Injectio | windows/remote/21757.txt
OmniHTTPd 1.1/2.0.x/2.4 - test.shtml Sample Application Cross-Site Scripting | windows/remote/21754.txt
OpenBSD HTTPd < 6.0 - Memory Exhaustion Denial of Service | openbsd/dos/41278.txt
Shellcodes: No Results
```

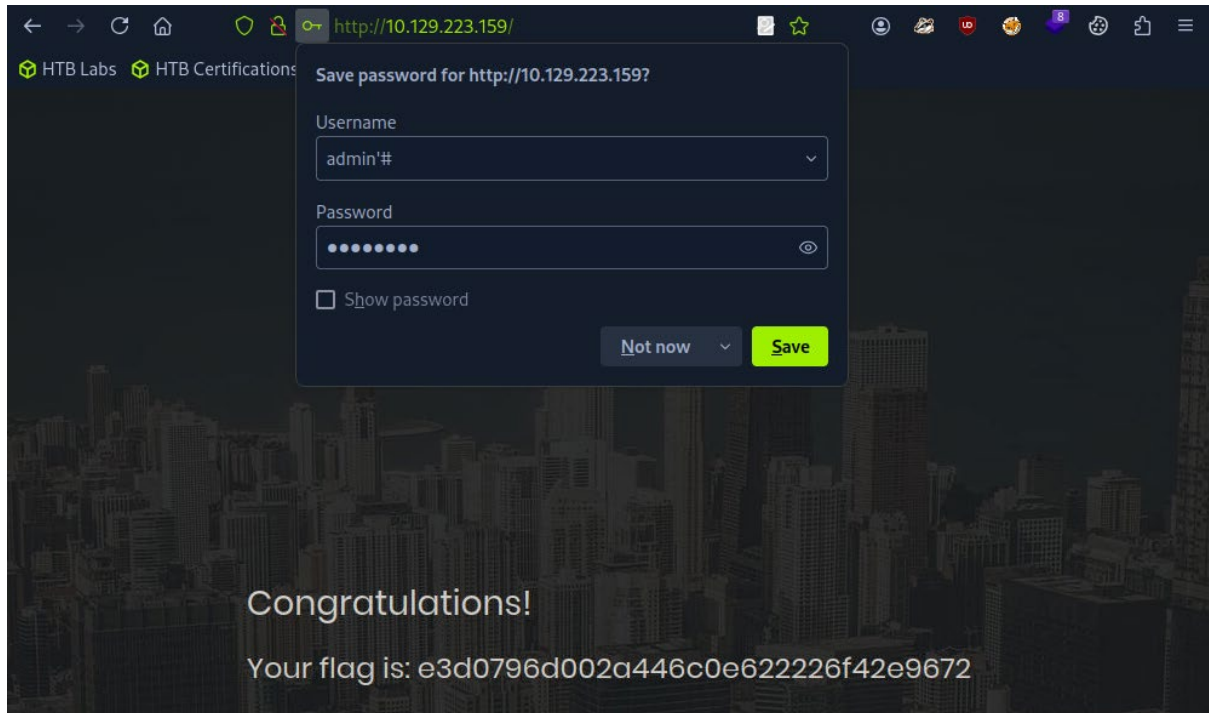
We know this service is running on port 80, so let's access it from our browser and try to login:

Attempt 1 Brute Force: username = admin, password= admin → Unsuccessful

Attempt 2 SQL Injection: username= ' OR 1'=1', password= irrelevant → Unsuccessful

Attempt 3 SQL Injection: username= admin, password = ' OR 1'=1' → Unsuccessful

Attempt 4 SQL Injection: username= admin'#, password= irrelevant → Successful



The reason this worked as an SQL injection is because the script input in the username field terminates the SQL injection query early and comments the rest of the query bypassing the password check. The dependency that the script relies on is the assumptions that a user named 'admin' exists.

Another machine down!

