Writeup: Spoof! - HackerDNA Cybersecurity Lab

Introduction

IP spoofing is a technique often used in network attacks to impersonate trusted sources, bypassing naive access controls that rely on client IP addresses. The *Spoof!* lab from HackerDNA demonstrates this vulnerability in a simple web application, where content is restricted to a specific "company network" IP. Available at HackerDNA, this easy challenge (great for beginners) simulates a tech solutions portal that checks incoming IPs but fails to validate them properly—allowing header manipulation to trick the server.

Objective: Start the lab VM, scan the target, identify the IP restriction on the web service, spoof your IP using curl to gain access, and retrieve the hidden flag.txt from the revealed path. This lab teaches the dangers of trusting unverified client-side data like IP headers.

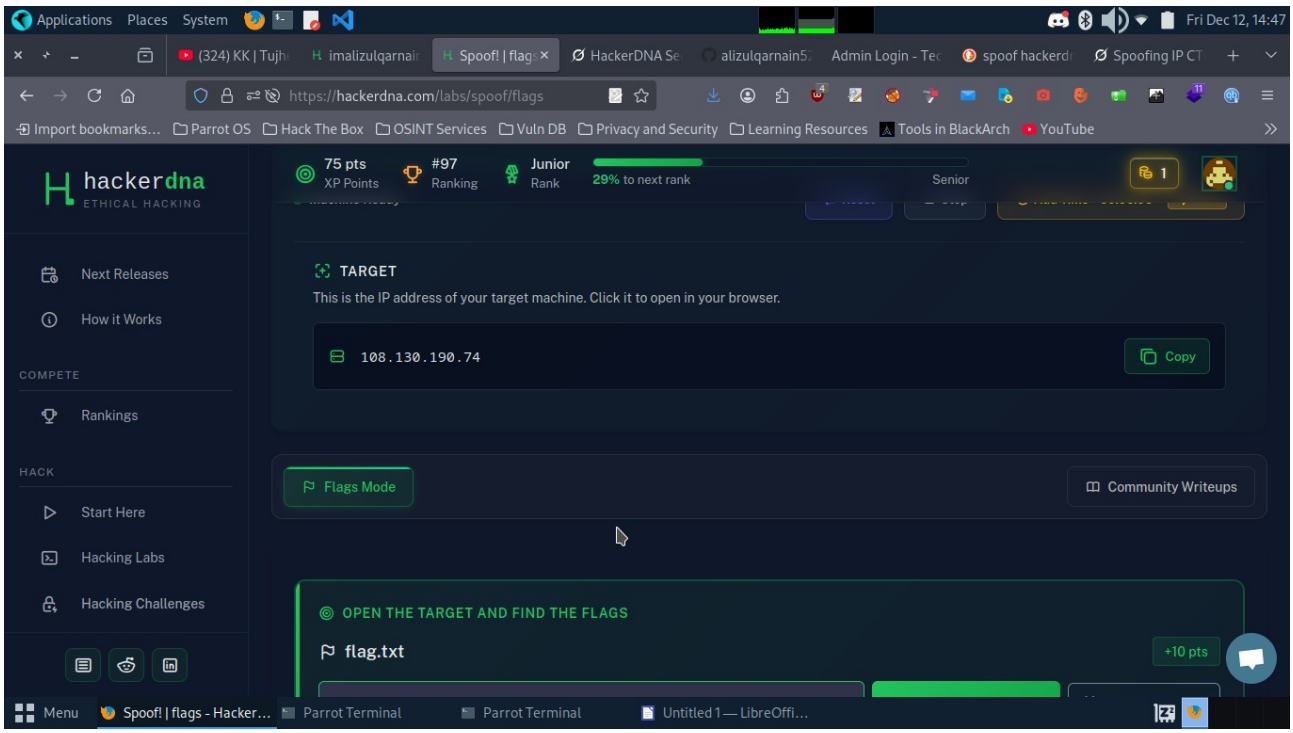
Difficulty: Easy
Points: 10 (1 flag worth +10 pts)
Success Rate: ~65% (as of December 2025)
Estimated Time: 5-10 minutes (plus 1-2 min setup)
Skills Tested: Network scanning with Nmap, HTTP header manipulation, IP spoofing basics, web reconnaissance

With HackerDNA's isolated instances, you get a safe playground to practice—highlighting why server-side IP checks should use more robust methods like VPNs or mutual TLS.

Lab Setup

1. **Access the Lab**: Log into HackerDNA Labs and search for "Spoof!" Hit "Start Lab" to spin up your private VM (up in ~1-2 minutes). This provides a dedicated environment with terminal access and a browser.
2. **Retrieve the Target IP**: The dashboard shows the target IP (e.g., 108.130.190.74), usually on port 80. Copy it—this is your target for scanning and exploitation. Keep the dashboard open for flag submission.

HackerDNA Lab Dashboard



Step 1: Scan the Target with Nmap

- In the terminal, run nmap ---open -Pn your_ip_address -vv
- Results: Only port 80/tcp open (HTTP, likely Apache/Nginx), no other services.

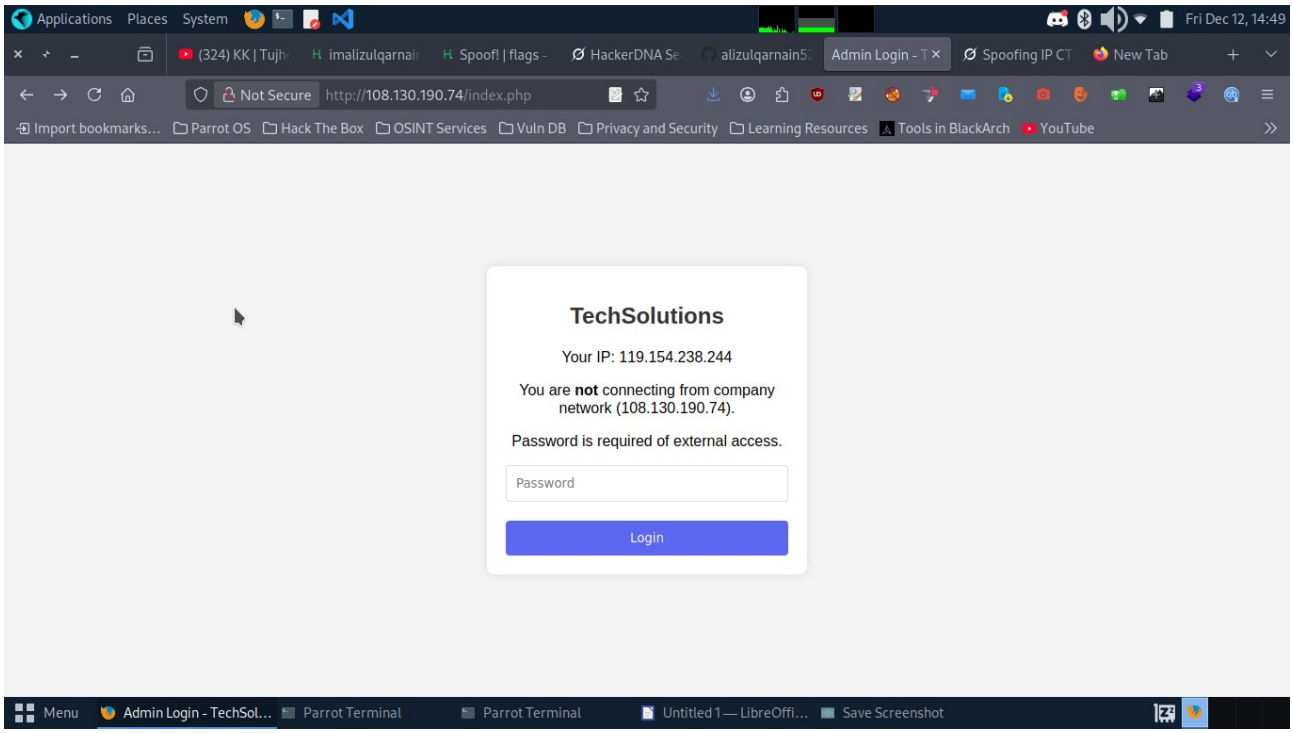
Nmap Scan:

```
┌──[unknown@parrot]─[~]
└──╼ $nmap --open -Pn 108.130.190.74 -vv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-12-12 14:24 EST
Initiating Parallel DNS resolution of 1 host. at 14:24
Completed Parallel DNS resolution of 1 host. at 14:24, 0.30s elapsed
Initiating Connect Scan at 14:24
Scanning ec2-108-130-190-74.eu-west-1.compute.amazonaws.com (108.130.190.74) [1000 ports]
Discovered open port 80/tcp on 108.130.190.74
Completed Connect Scan at 14:25, 16.56s elapsed (1000 total ports)
Nmap scan report for ec2-108-130-190-74.eu-west-1.compute.amazonaws.com (108.130.190.74)
Host is up, received user-set (0.17s latency).
Scanned at 2025-12-12 14:24:50 EST for 17s
Not shown: 999 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 16.91 seconds
```

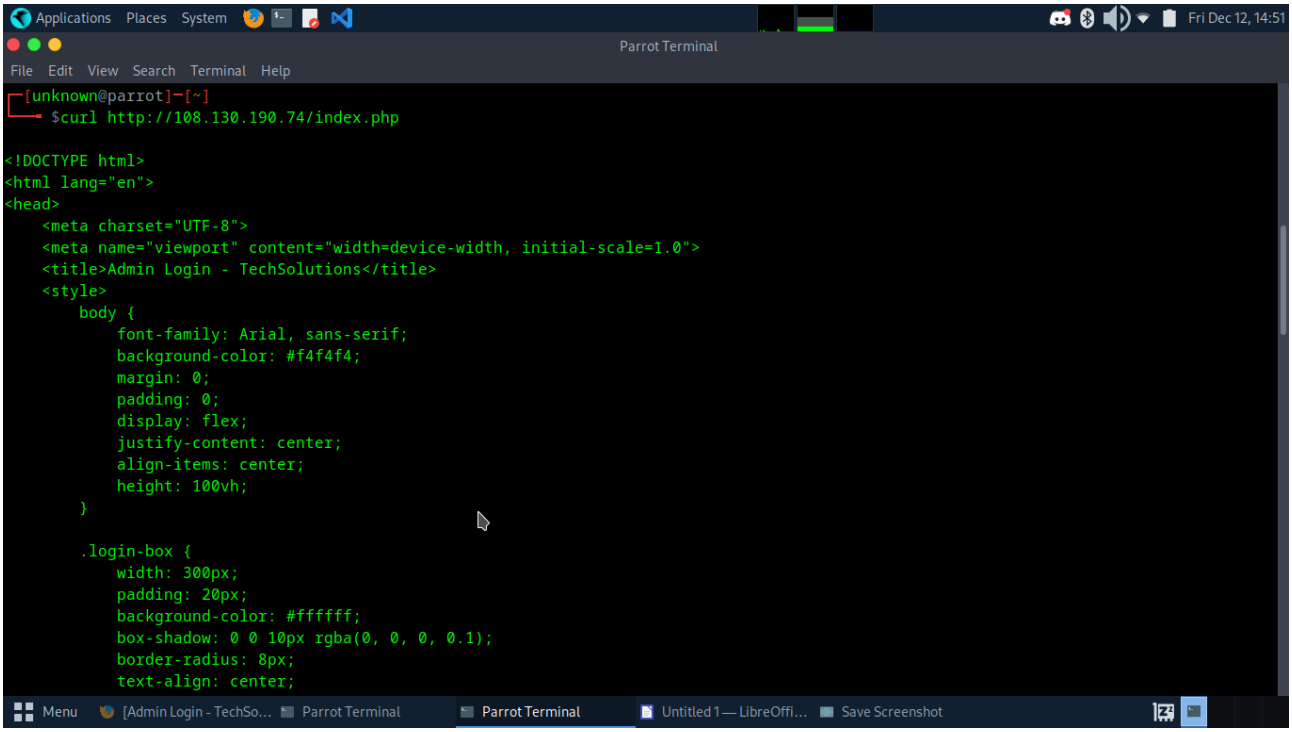
Step 2: Access the Web Page

- Open http://<target_IP>:80 in your browser.
- The page loads a "TechSolutions" login portal: "Your IP: <your_IP> You are not connecting from company network (108.130.190.74)." It demands a password for internal access—clear IP restriction.

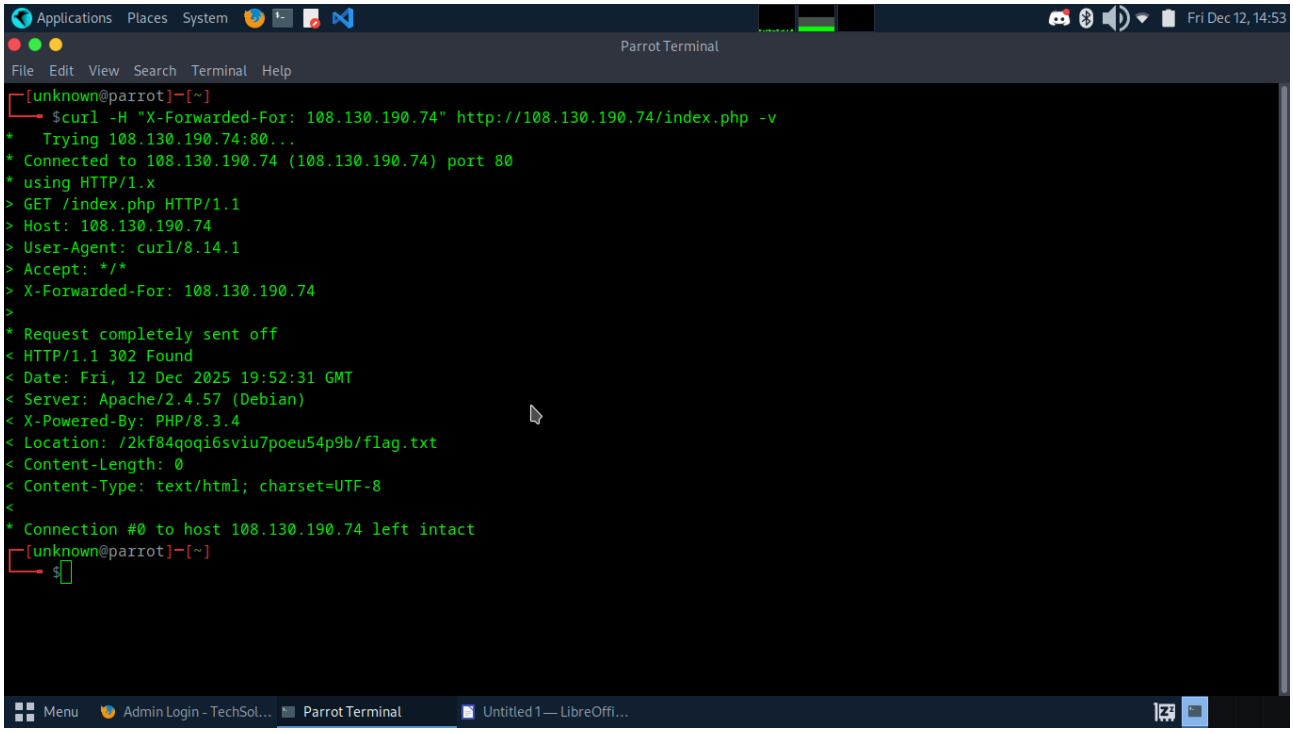


Step 3: Spoof the IP with Curl

- The restriction likely checks headers like X-Forwarded-For. Spoof it: curl http://<target_IP> -H "X-Forwarded-For: 108.130.190.74".
- Response: Page source or content now accessible, revealing the flag location

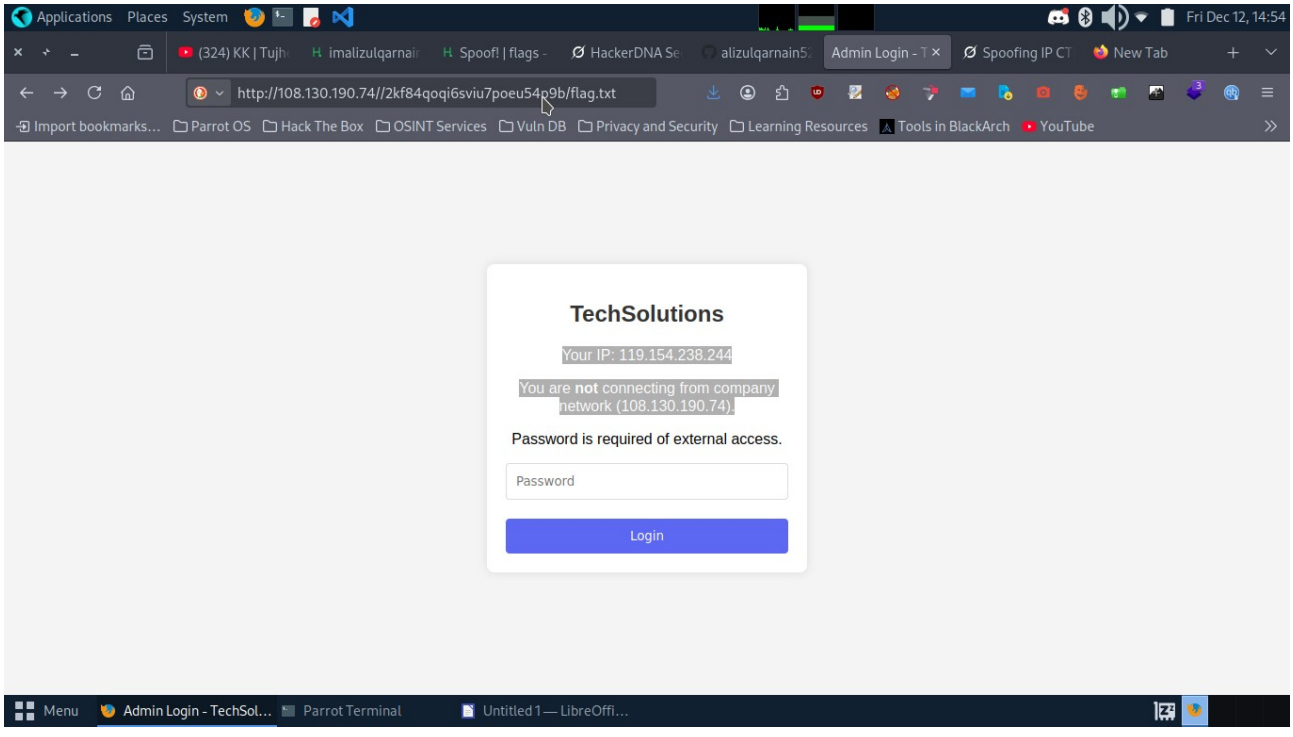


then using curl we'll soof ip address



Step 4: Retrieve the Flag

- Copy the revealed path (e.g., /flag.txt).
- Access it: http://<target_IP>/flag.txt (or use curl again with the spoofed header if required).
- The flag displays directly—submit it via the dashboard.



So after hitting enter congrats we have flag and now submit the flag.

Happy spoofing!