WEB REQUESTS

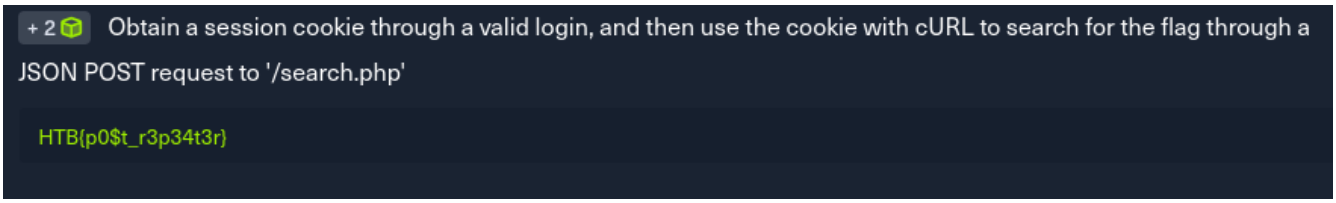
INTRODUCTION

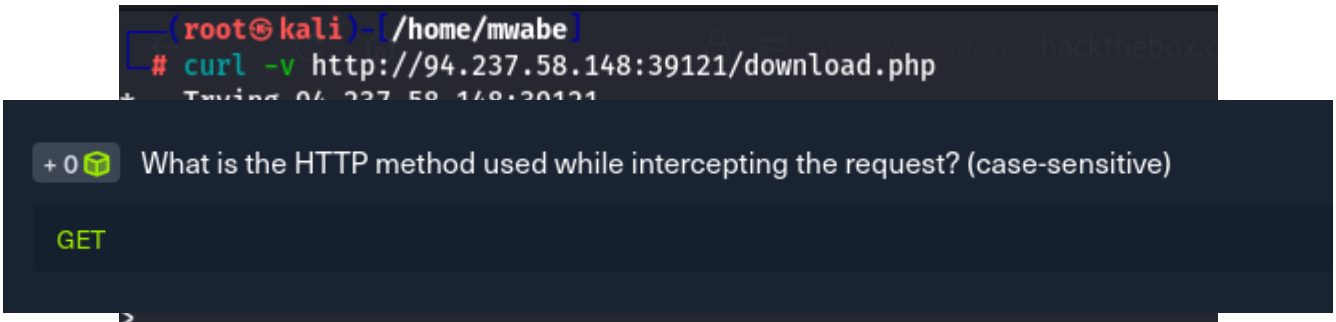
In this module I got to cover the following ares majorly:

- HTTP
- HTTPS(secure version of http)
- HTTP requests and responses
- HTTP methods.:GET = used to request resource from a server
  POST = used to upload data to the server
  PUT = used to update data on database
  DELETE = used to delete data from the server

I also learnt how to use the command line url(cURL) to achieve the above from my terminal.

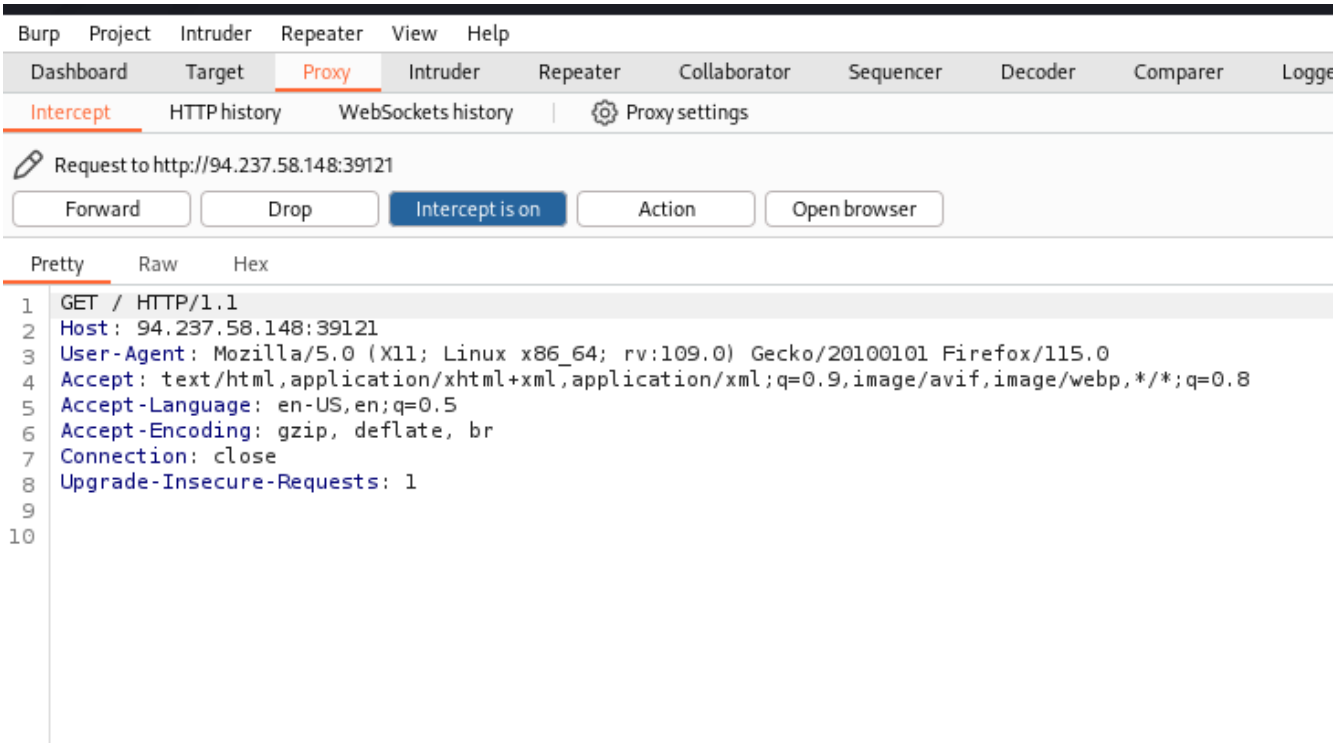
Q & A

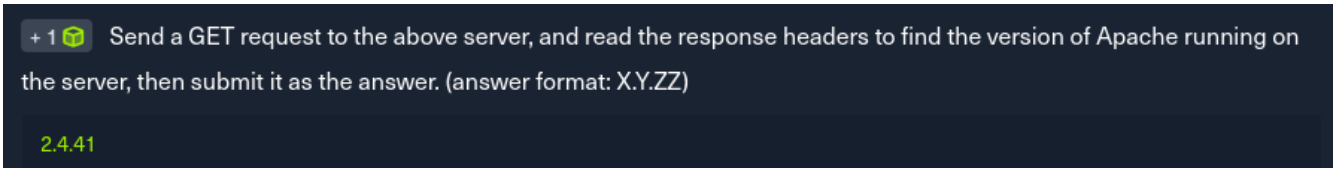




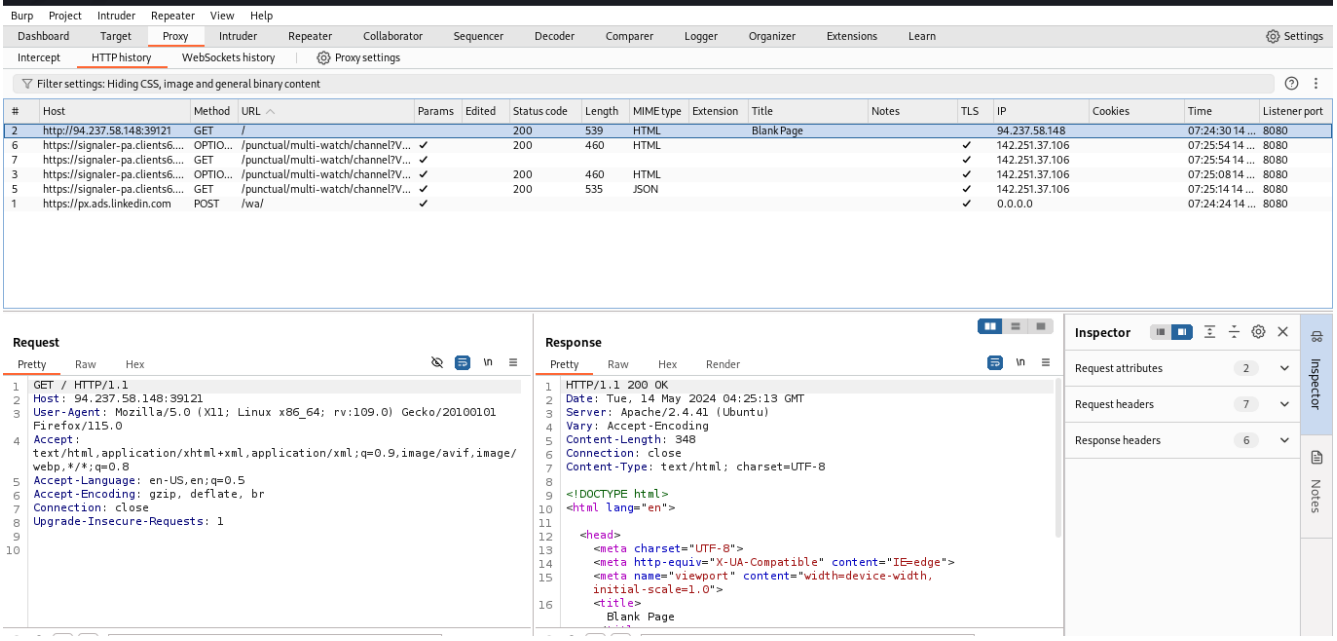

```
┌──(root㉿kali)-[/home/mwabe]
└─# curl -v http://94.237.58.148:39121/download.php
*   Trying 94.237.58.148:39121...
* Connected to 94.237.58.148 (94.237.58.148) port 39121 (#0)
> GET /download.php HTTP/1.1
> Host: 94.237.58.148:39121
> User-Agent: curl/8.7.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 14 May 2024 04:12:16 GMT
< Server: Apache/2.4.41 (Ubuntu)
< Content-Description: File Transfer
< Cache-Control: no-cache, must-revalidate
< Expires: 0
< Content-Disposition: attachment; filename="flag.txt"
< Content-Length: 20
< Pragma: public
< Content-Type: text
<
* Connection #0 to host 94.237.58.148 left intact
HTB{64$!c_cURL_u$3r}

┌──(root㉿kali)-[/home/mwabe]
└─#
```





so I intercepted the request using the burp tool, sent it to the repeater and there got the response in which information about the version of the Apache server is just as shown below.

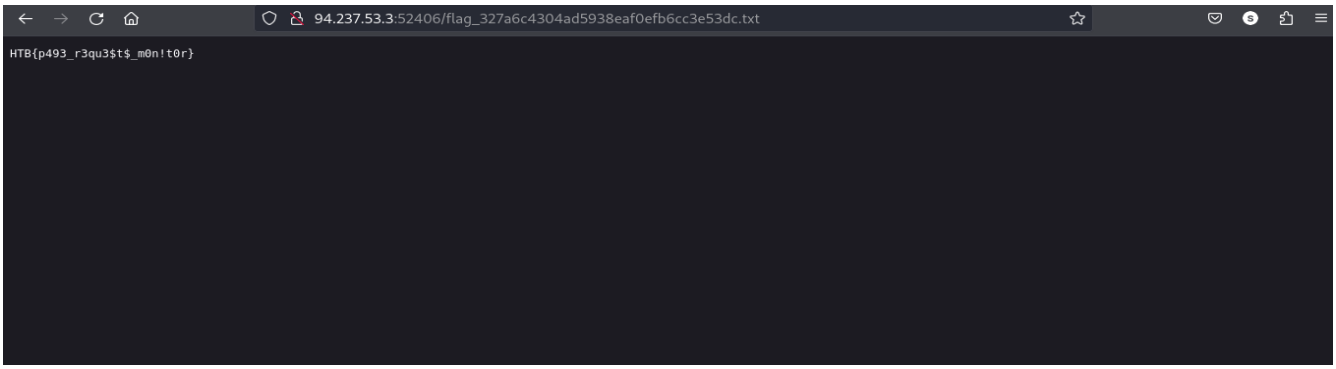


+2 The server above loads the flag after the page is loaded. Use the Network tab in the browser devtools to see what requests are made by the page, and find the request to the flag.

HTB{p493_r3qu3t_m0n!t0r}

The images below describes the approaches I took to find the flag

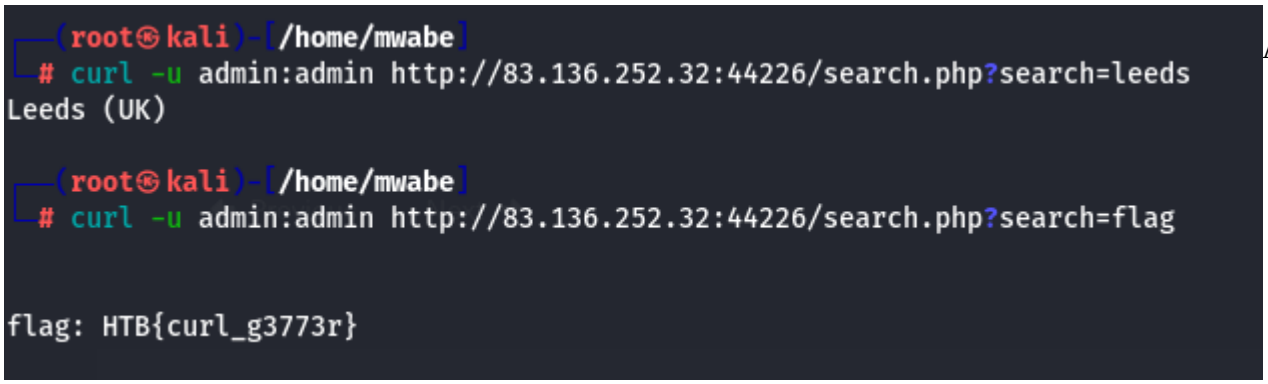
after finding the path to the flag from the network tab from our devtool I added it to the url path and here was the result



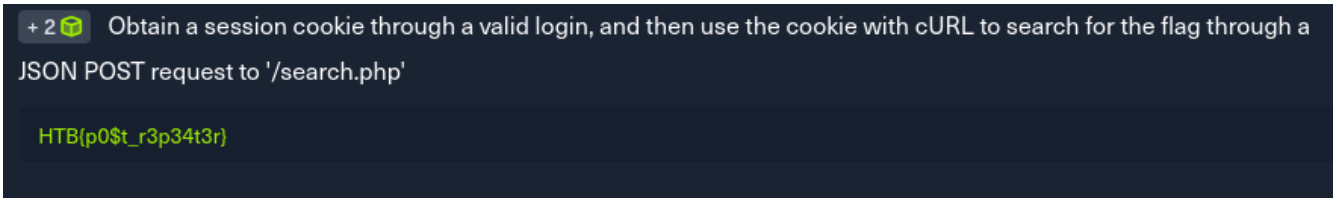
[

+2 The exercise above seems to be broken, as it returns incorrect results. Use the browser devtools to see what is the request it is sending when we search, and use cURL to search for 'flag' and obtain the flag.

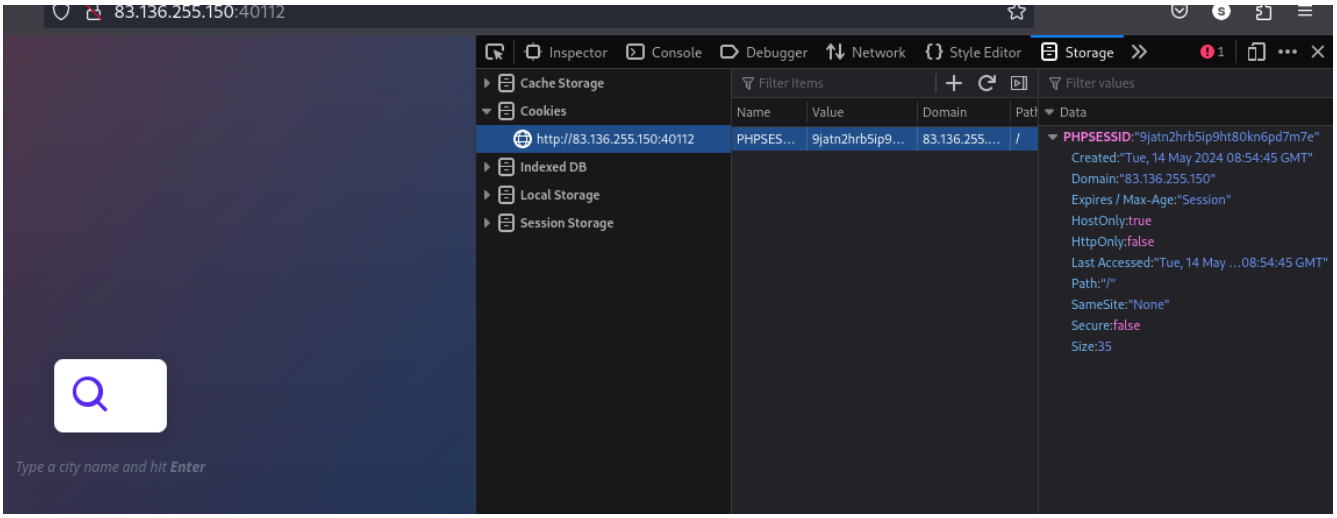
HTB{curl_g3773r}



As shown in the image above, I authenticated as the admin using the admin credentials the at the search parameter I used the value “leeds” and their the city was displayed on the terminal. So in the second command I replace leeds with flag and boom! The flag was displayed.



After authenticating myself, a cookie was returned and by navigating to the devtools at the nework or storage tab, I got the authentication cookie.



by use of authentication cookie, the search parameter and the head in combination with the curl cmd, I managed to retrieve the flag as shown in the image below.

```
┌──(root㉿kali)-[/home/mwabe]
└─# curl -X POST -d '{"search":"london"}' -b 'PHPSESSID=9jatn2hrb5ip9ht80kn6pd7m7e' -H 'Content-Type: application/json' http://83.136.255.150:40112/search.php
["London (UK)"]

┌──(root㉿kali)-[/home/mwabe]
└─# curl -X POST -d '{"search":"flag"}' -b 'PHPSESSID=9jatn2hrb5ip9ht80kn6pd7m7e' -H 'Content-Type: application/json' http://83.136.255.150:40112/search.php
["flag: HTB{p0$t_r3p34t3r}"]

┌──(root㉿kali)-[/home/mwabe]
└─#
```

+2 First, try to update any city's name to be 'flag'. Then, delete any city. Once done, search for a city named 'flag' to get the flag.

HTB{crud_4p!_m4n!pul4t0r}

So in this case, I had first to add a city in the database then delete and retrieved the flag. Here are images that explain how I achieved that.

```
┌──(root㉿kali)-[/home/mwabe]
└─# curl -X PUT http://94.237.49.166:37792/api.php/city/london -d '{"city_name":"flag", "country_name":"UK"}' -H 'Content-Type: application/json'


┌──(root㉿kali)-[/home/mwabe]
└─# curl -s http://94.237.49.166:37792/api.php/city/ | jq

[
  {
    "city_name": "flag",
    "country_name": "UK"
  },
  {
    "city_name": "Birmingham",
    "country_name": "(UK)"
  },
  {
    "city_name": "Leeds",
    "country_name": "(UK)"
```

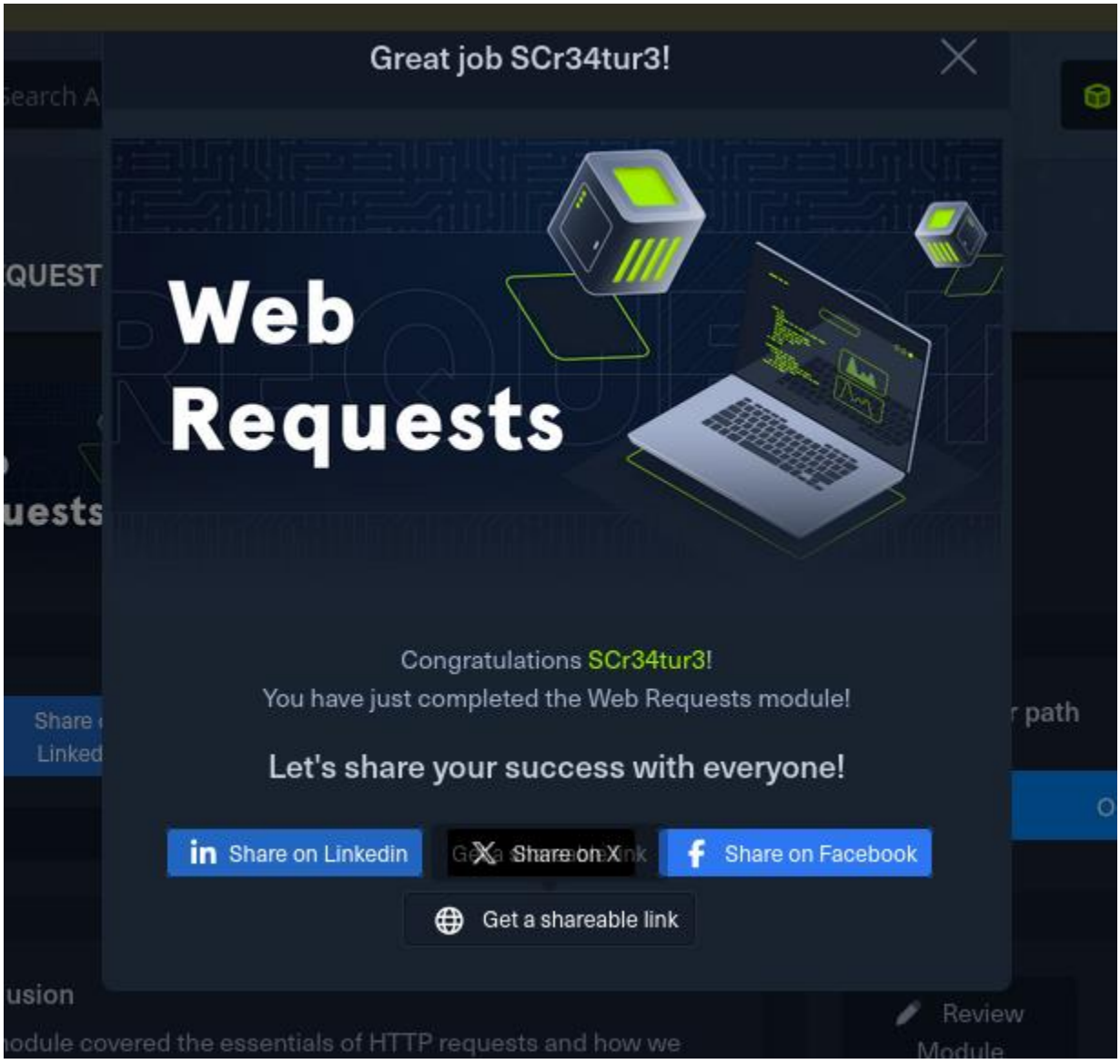
The command above enabled me to update a city called flag under a country called UK. And as you can see our city and country_name were successfully updated as shown in the image above.

```
┌──(root㉿kali)-[/home/mwabe]
└─# curl -X DELETE http://94.237.49.166:37792/api.php/city/Leeds


┌──(root㉿kali)-[/home/mwabe]
└─# curl -s http://94.237.49.166:37792/api.php/city/ | jq

[
  {
    "city_name": "flag",
    "country_name": "HTB{crud_4p!_m4n!pul4t0r}"
  },
  {
    "city_name": "Birmingham",
    "country_name": "(UK)"
  },
  {
    "city_name": "Glasgow",
    "country_name": "(UK)"
```

The image above displays how I deleted a city from the database and successfully retrieved the flag by displaying all the contents in the database in json format.



https://academy.hackthebox.com/achievement/1287818/35

CONCLUSION

In conlusion, understanding how web requests works in combination with a strong foundation in deciphering how web application operates puts us to edge when I comes to performing pentesting and handling various security potholes into systems, network and applications.