HTB Academy- Web Requests

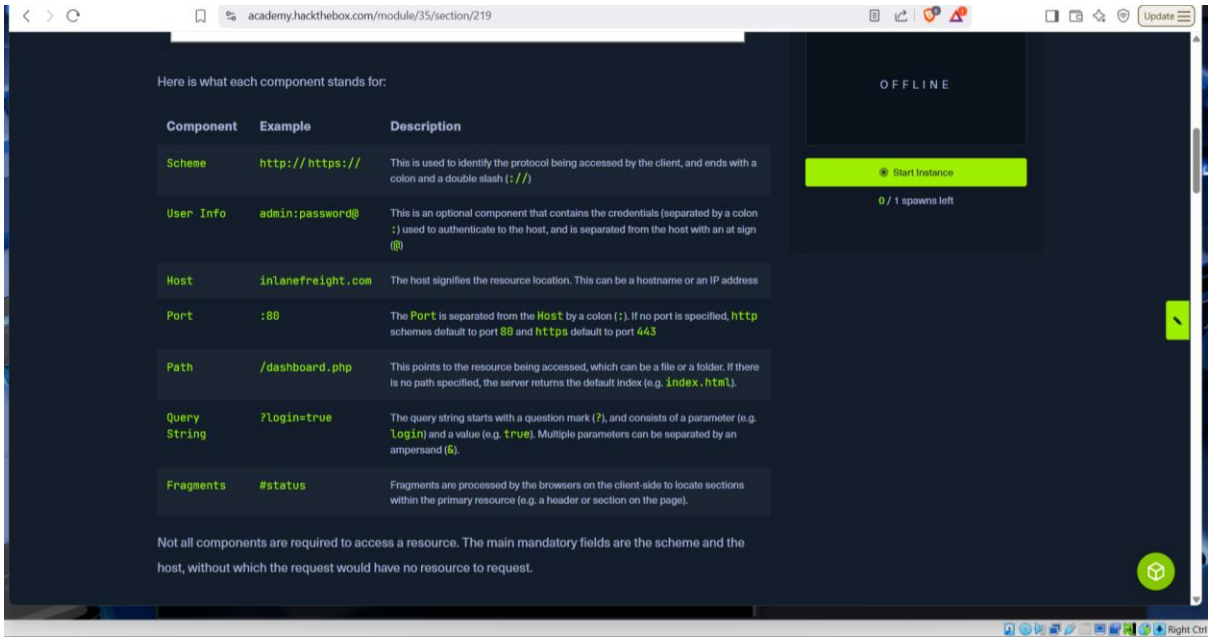
Introduction

This report explores fundamental concepts related to HTTP and web requests as covered in the HTB Academy module. It focuses on the structure and use of HTTP and HTTPS protocols, the behavior of different HTTP methods, and how to inspect requests and responses using tools such as curl and browser developer tools. The tasks provided in the module aim to reinforce an understanding of web communication, especially from a security testing perspective. By engaging in practical exercises, I was able to apply and analyze HTTP behaviors that are commonly encountered in real-world web environments.

HyperText Transfer Protocol (HTTP)

In this Section I covered the Components of a URL, http flow and the Curl Command.

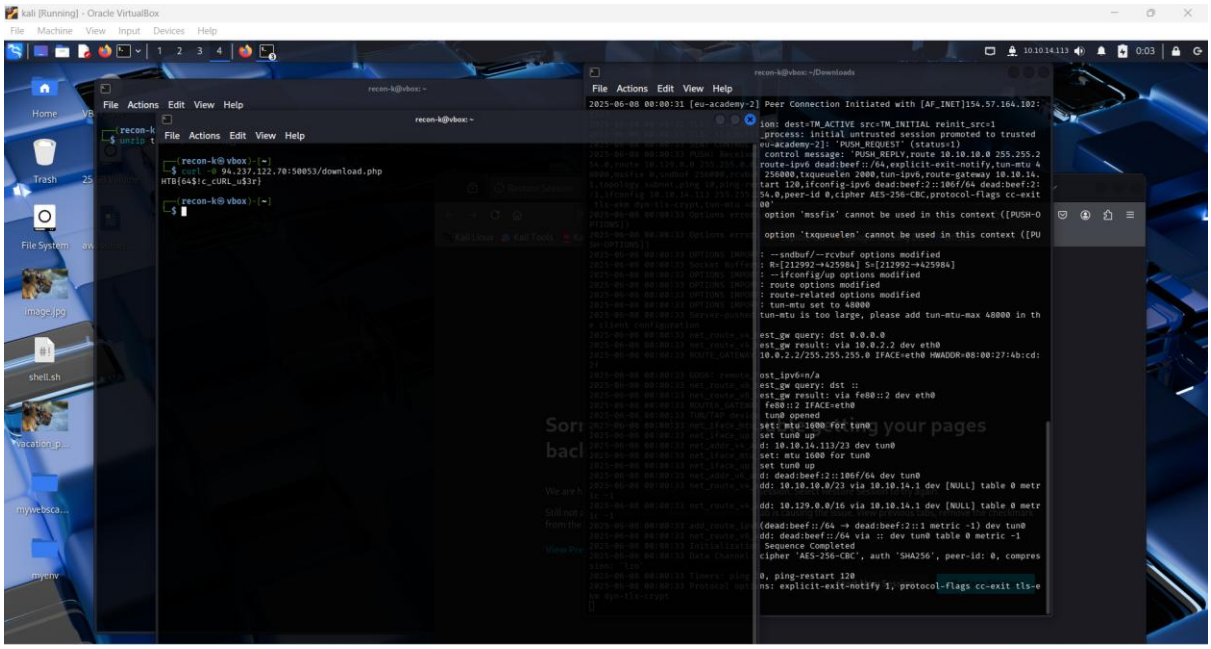
The curl -O flag is used to download a page or a file



Questions:

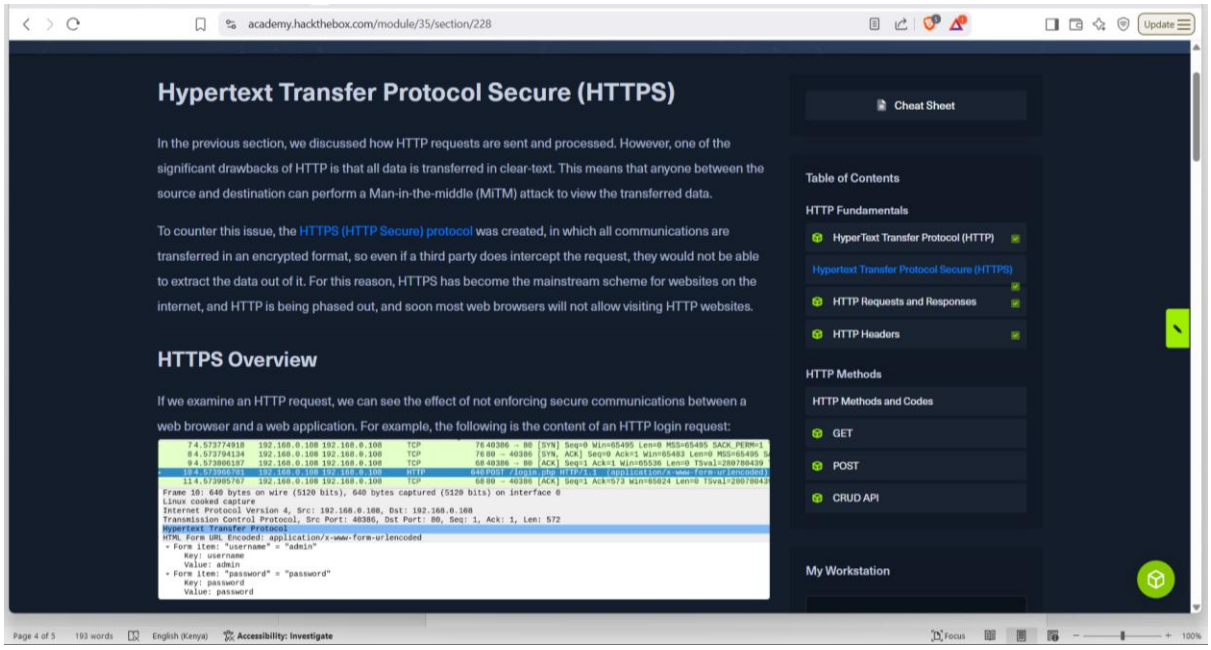
To get the flag, start the above exercise, then use cURL to download the file returned by '/download.php' in the server shown above

Flag: HTB{64$!c_cURL_u$3r}



Hypertext Transfer Protocol Secure (HTTPs)

This was created to counter Man in the Middle attacks as most encryption in this case are encrypted.

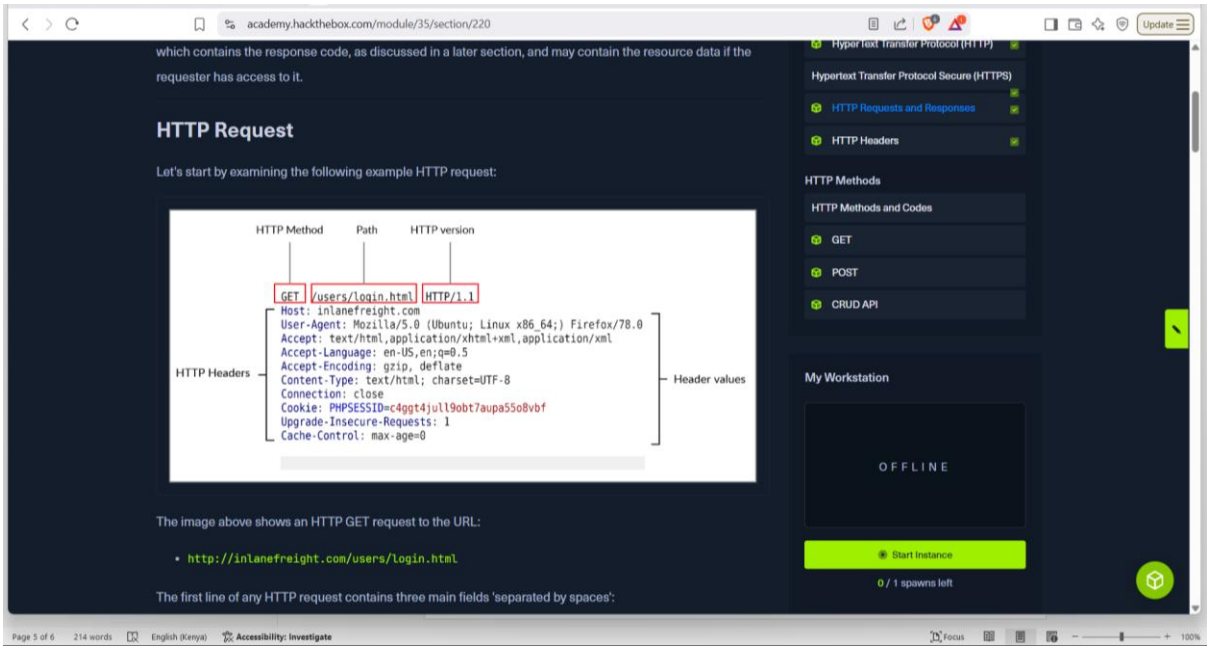


HTTPS FLOW:

The -k flag on curl command can be used to ignore outdated SSL certificates as curl by default supports https.

HTTP Requests and Responses

I started by understanding the different parts of a Http requests:



Explanation:

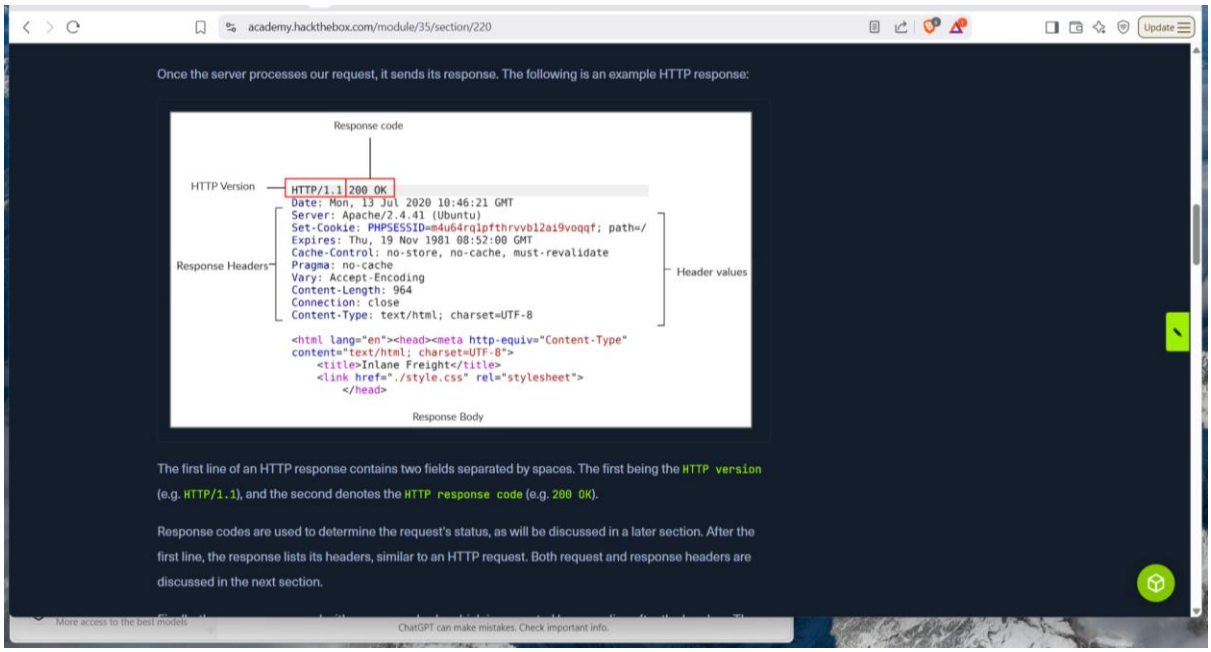
The first line of any HTTP request contains three main fields 'separated by spaces':

Field	Example	Description
`Method`	`GET`	The HTTP method or verb, which specifies the type of action to perform.
`Path`	`/users/login.html`	The path to the resource being accessed. This field can also be suffixed with a query string (e.g. `?username=user`).
`Version`	`HTTP/1.1`	The third and final field is used to denote the HTTP version.

The next set of lines contain HTTP header value pairs, like `Host`, `User-Agent`, `Cookie`, and many other possible headers. These headers are used to specify various attributes of a request. The headers are terminated with a new line, which is necessary for the server to validate the request. Finally, a request may end with the request body and data.

Note: HTTP version 1.X sends requests as clear-text, and uses a new-line character to separate different fields and different requests. HTTP version 2.X, on the other hand, sends requests as binary data in a dictionary form.

I then proceeded to http response



Explanation:

The first line of an HTTP response contains two fields separated by spaces. The first being the `HTTP version` (e.g. `HTTP/1.1`), and the second denotes the `HTTP response code` (e.g. `200 OK`).

Response codes are used to determine the request's status, as will be discussed in a later section. After the first line, the response lists its headers, similar to an HTTP request. Both request and response headers are discussed in the next section.

Finally, the response may end with a response body, which is separated by a new line after the headers. The response body is usually defined as `HTML` code. However, it can also respond with other code types such as `JSON`, website resources such as images, style sheets or scripts, or even a document such as a PDF document hosted on the webserver.

To view the full HTTP request and response, we can simply add the -v verbose flag on curl.

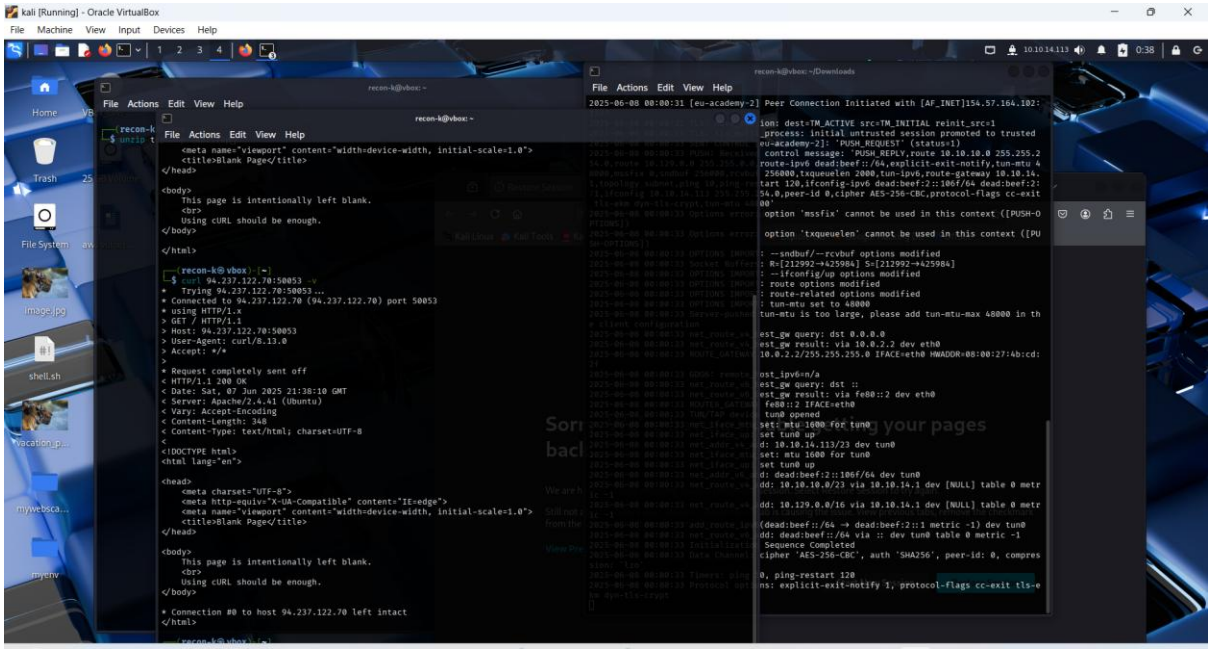
Browser dev tools can also be used for web assessment during pentesting.

To open the browser devtools in either Chrome or Firefox, we can click [CTRL+SHIFT+I] or simply click [F12]. The devtools contain multiple tabs, each of which has its own use

Questions

What is the HTTP method used while intercepting the request? GET

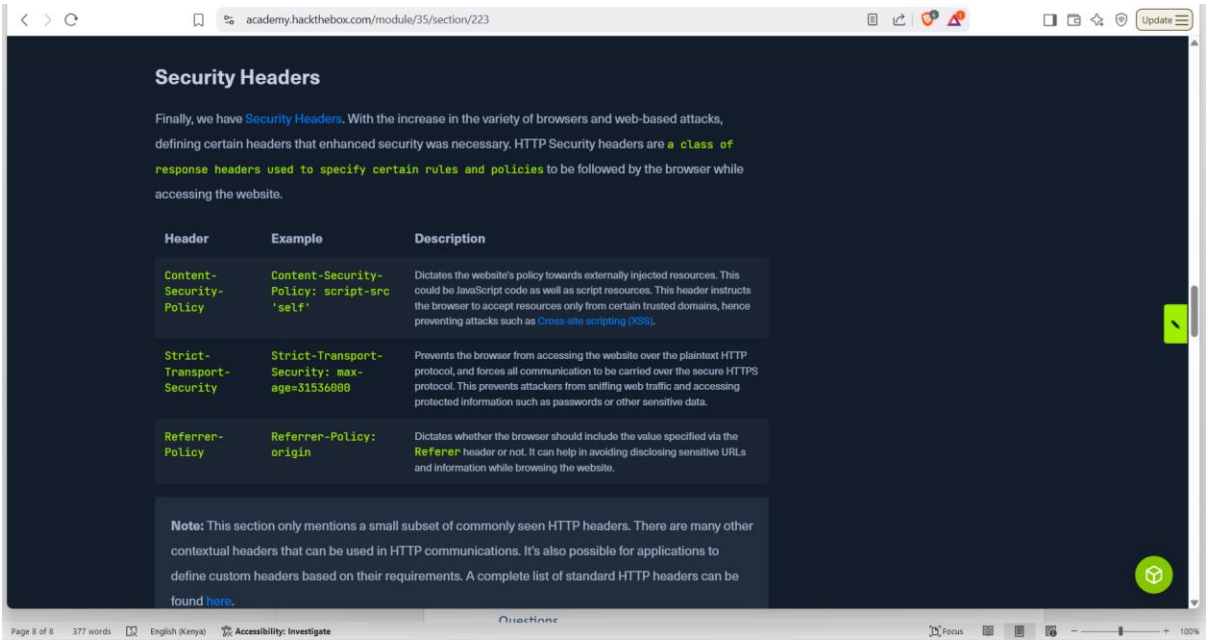
Send a GET request to the above server, and read the response headers to find the version of Apache running on the server, then submit it as the answer. (answer format: X.Y.ZZ)
2.4.41



HTTP Headers

Headers can have one or multiple values, appended after the header name and separated by a colon. We can divide headers into the following categories:

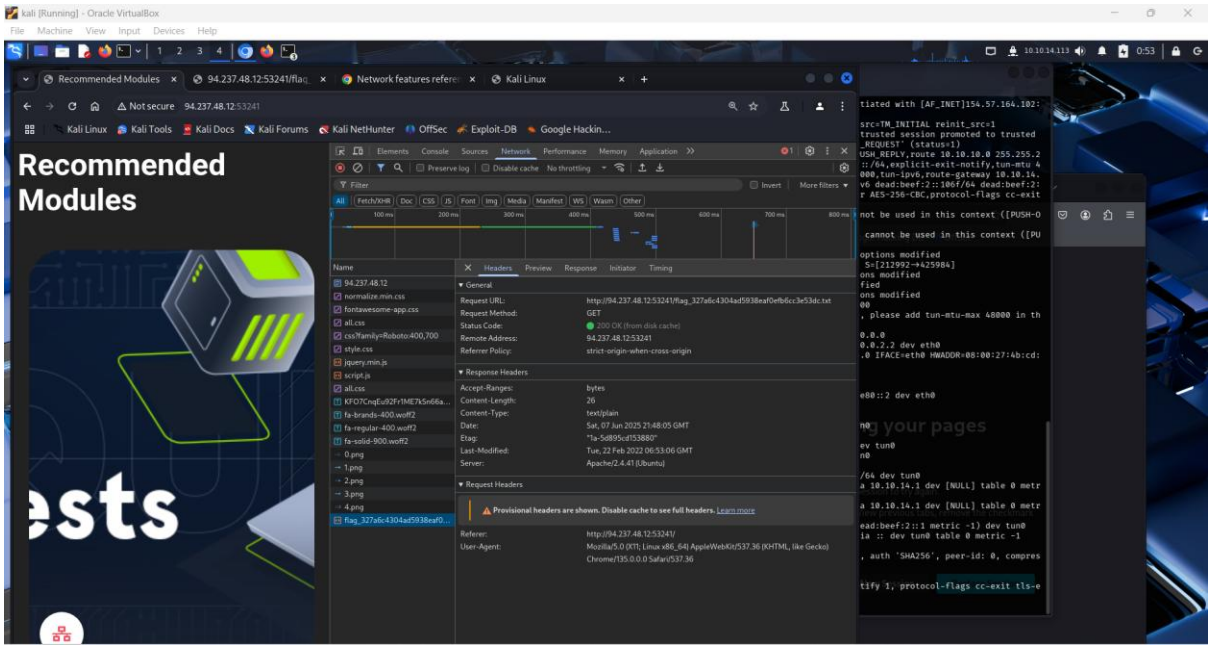
1. General Headers
2. Entity Headers
3. Request Headers
4. Response Headers
5. Security Headers

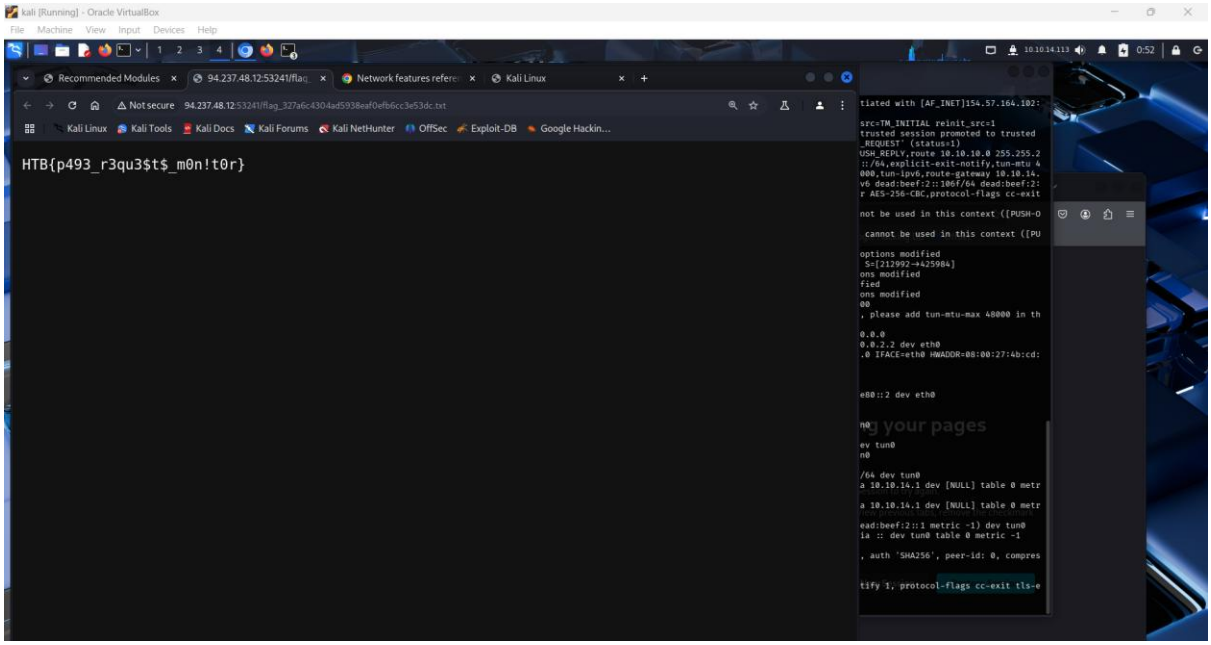


In curl, If we were only interested in seeing the response headers, then we can use the -I flag to send a HEAD request and only display the response headers. Furthermore, we can use the -i flag to display both the headers and the response body (e.g. HTML code). The difference between the two is that -I sends a HEAD request (as will see in the next section), while -i sends any request we specify and prints the headers as well.

Questions

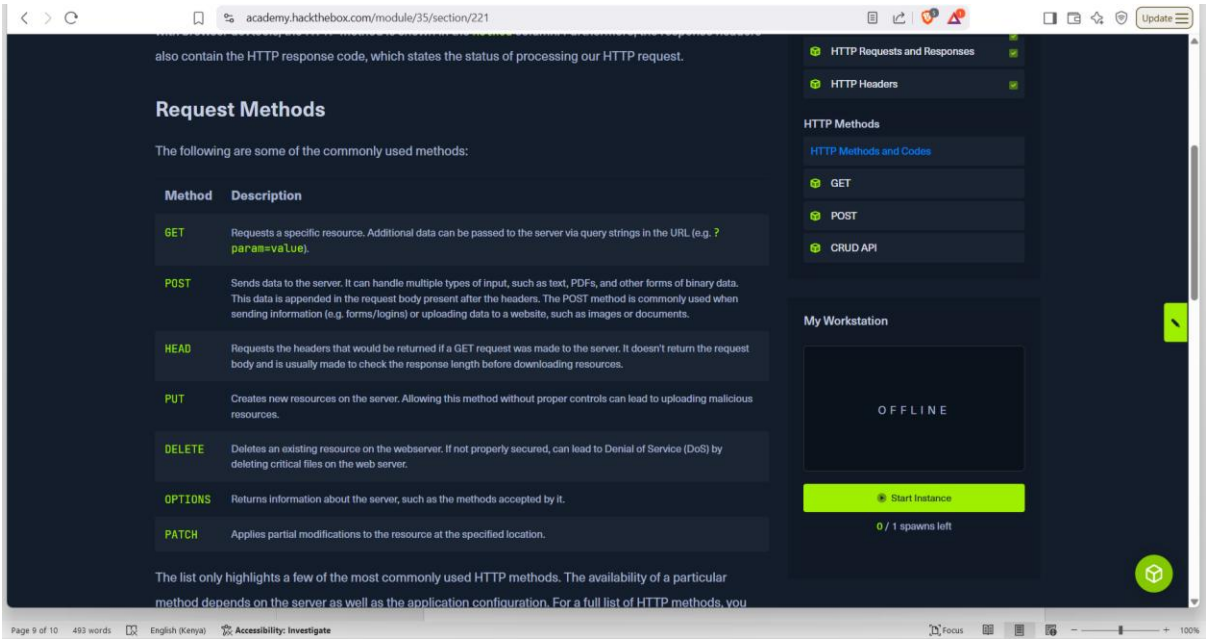
The server above loads the flag after the page is loaded. Use the Network tab in the browser devtools to see what requests are made by the page, and find the request to the flag.



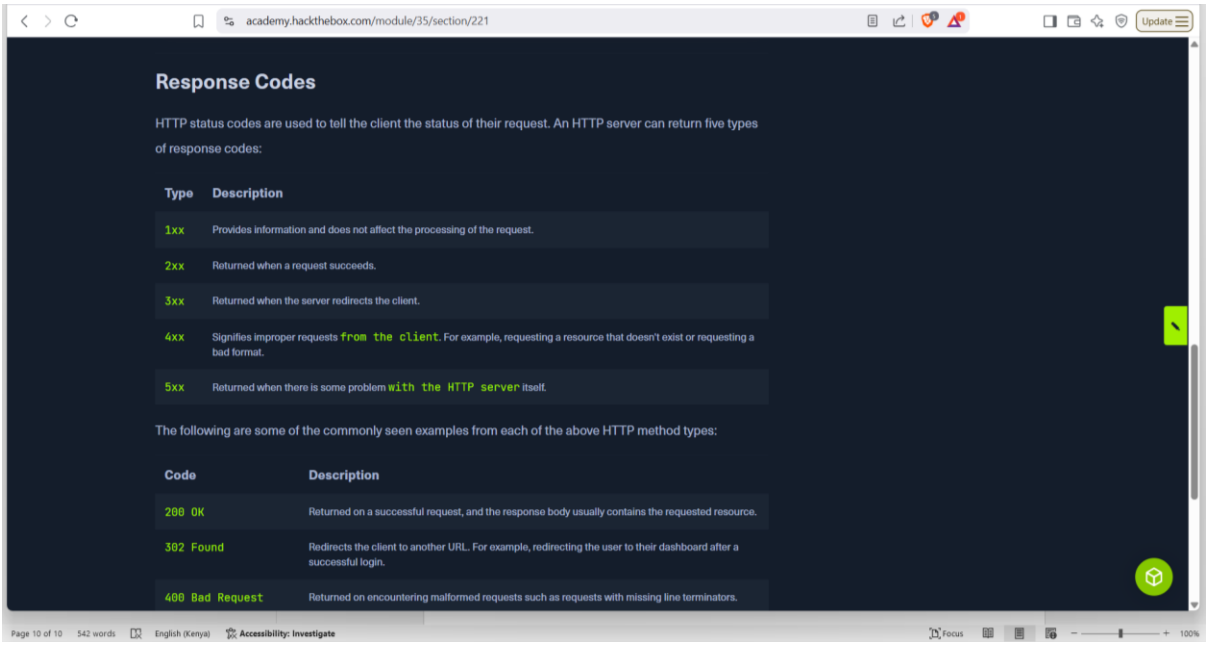


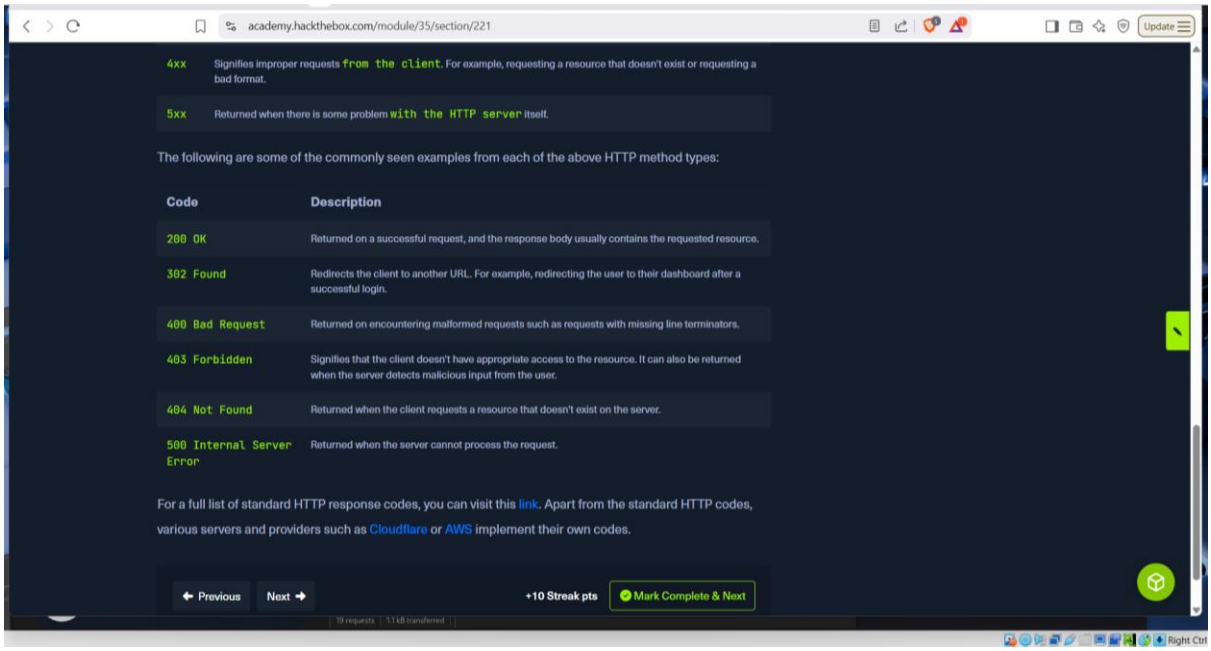
HTTP Methods and Codes

HTTP supports multiple methods for accessing a resource. In the HTTP protocol, several request methods allow the browser to send information, forms, or files to the server. These methods are used, among other things, to tell the server how to process the request we send and how to reply.



Response codes

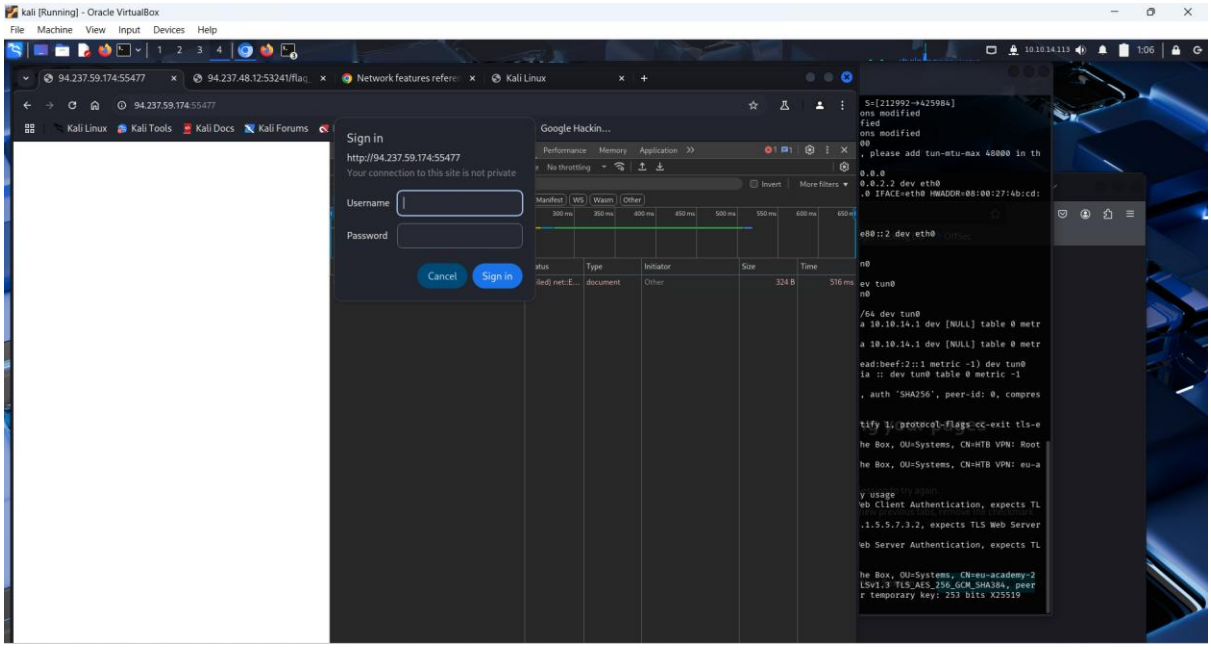




GET

Whenever we visit any URL, our browsers default to a GET request to obtain the remote resources hosted at that URL. Once the browser receives the initial page it is requesting; it may send other requests using various HTTP methods.

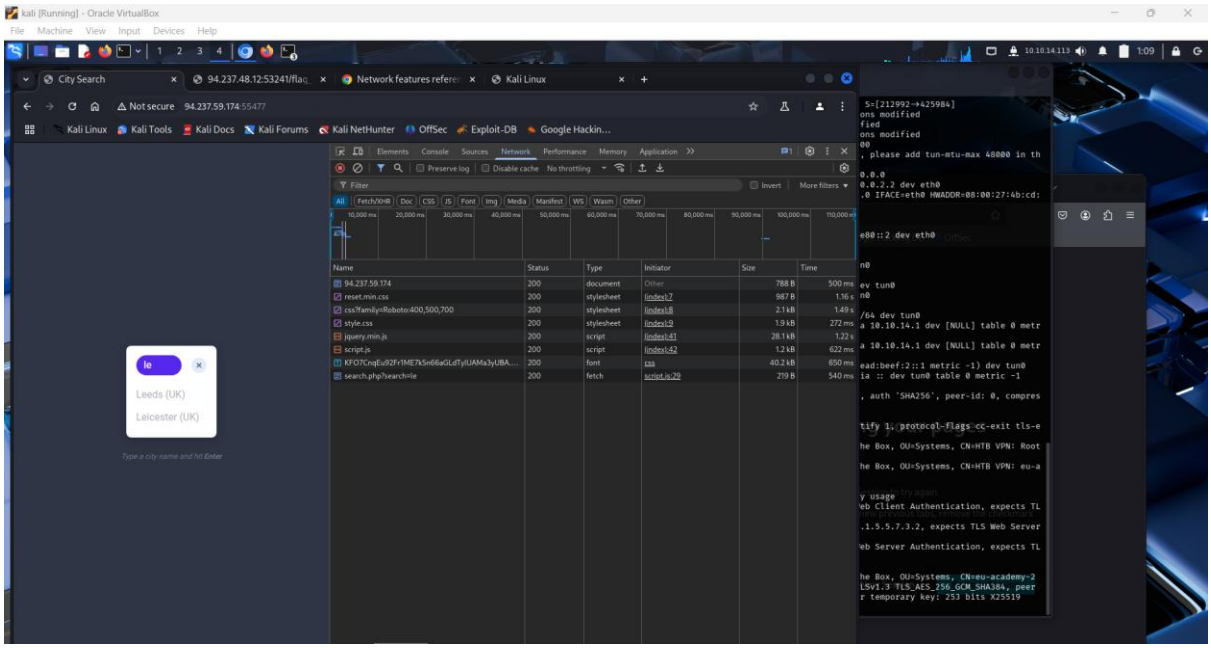
HTTP basic Auth:



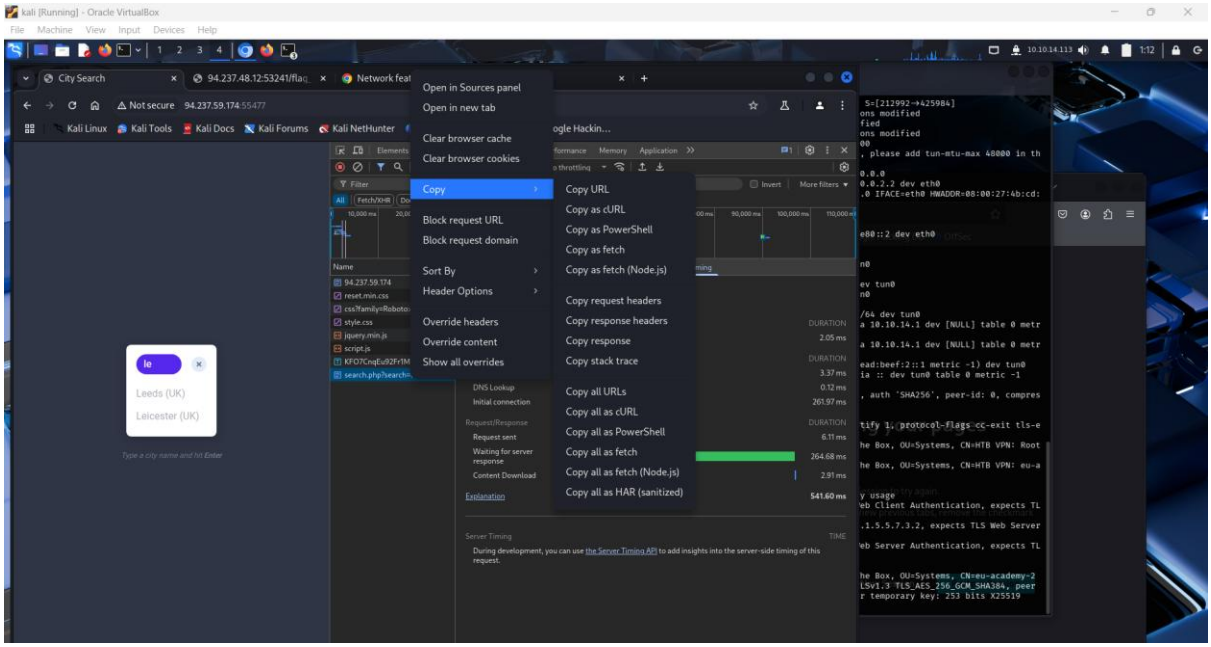
Question:

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}

To do this i entered the ip and port number and went to the developer tools in the network tab

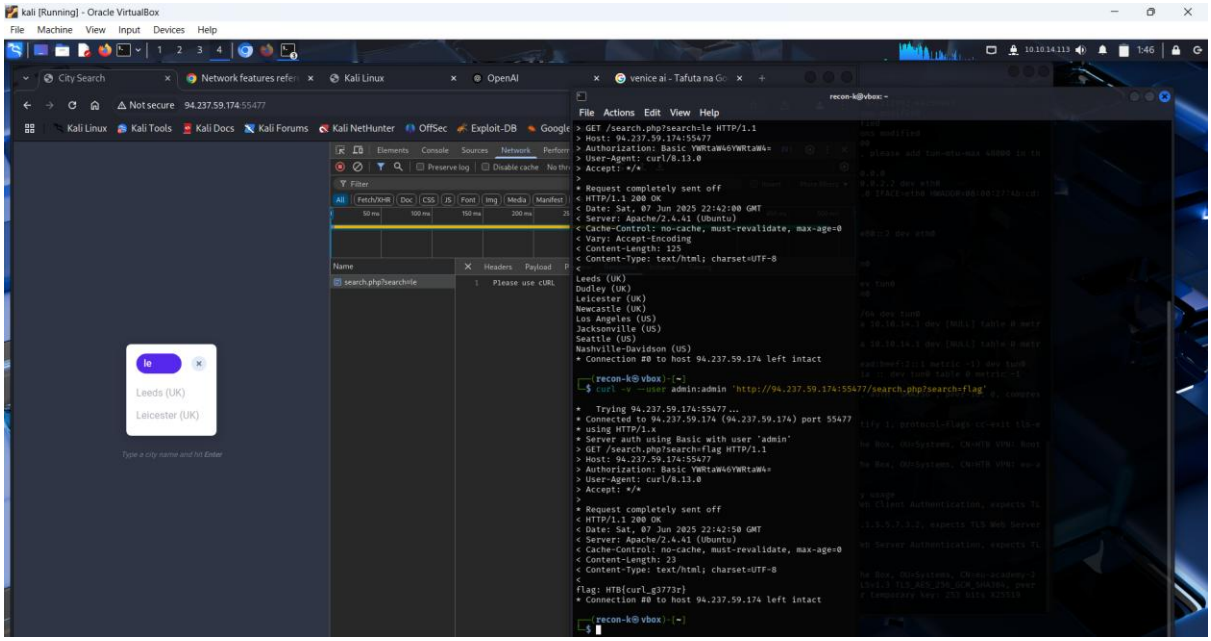


Authenticated and searched “le”, then copied the fetch request using curl



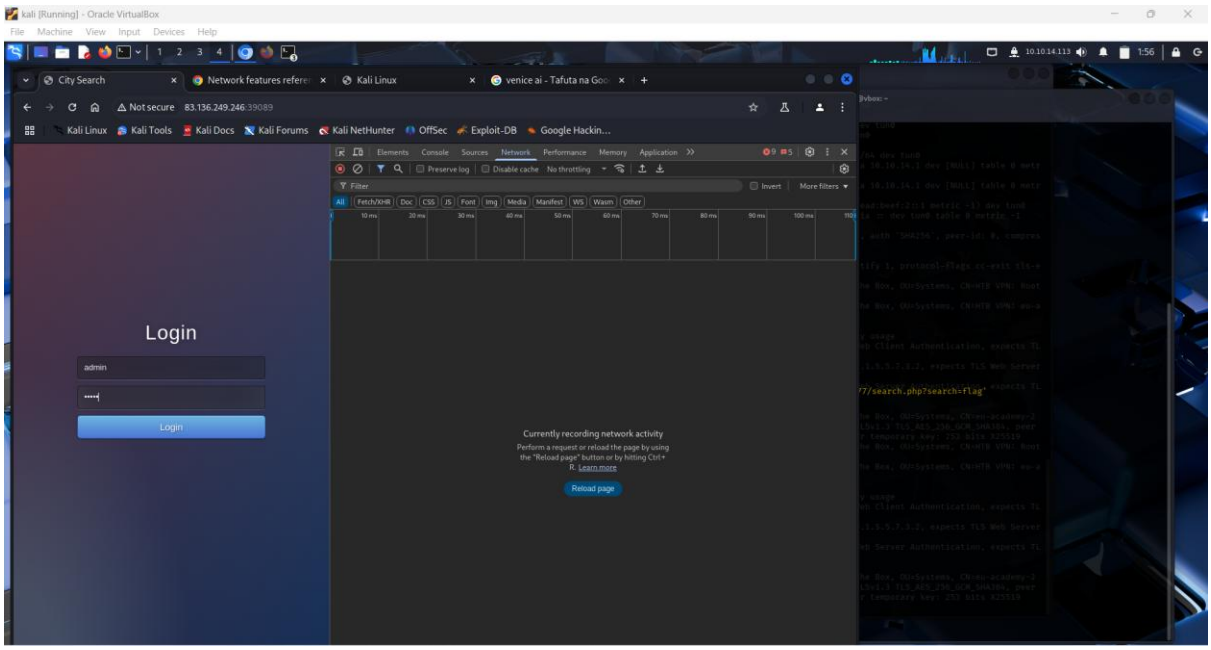
After several failed attempts using the copied curl command from the dev tools i used the basic command with authentication

curl -v --user admin:admin 'http://94.237.59.174:55477/search.php?search=le'



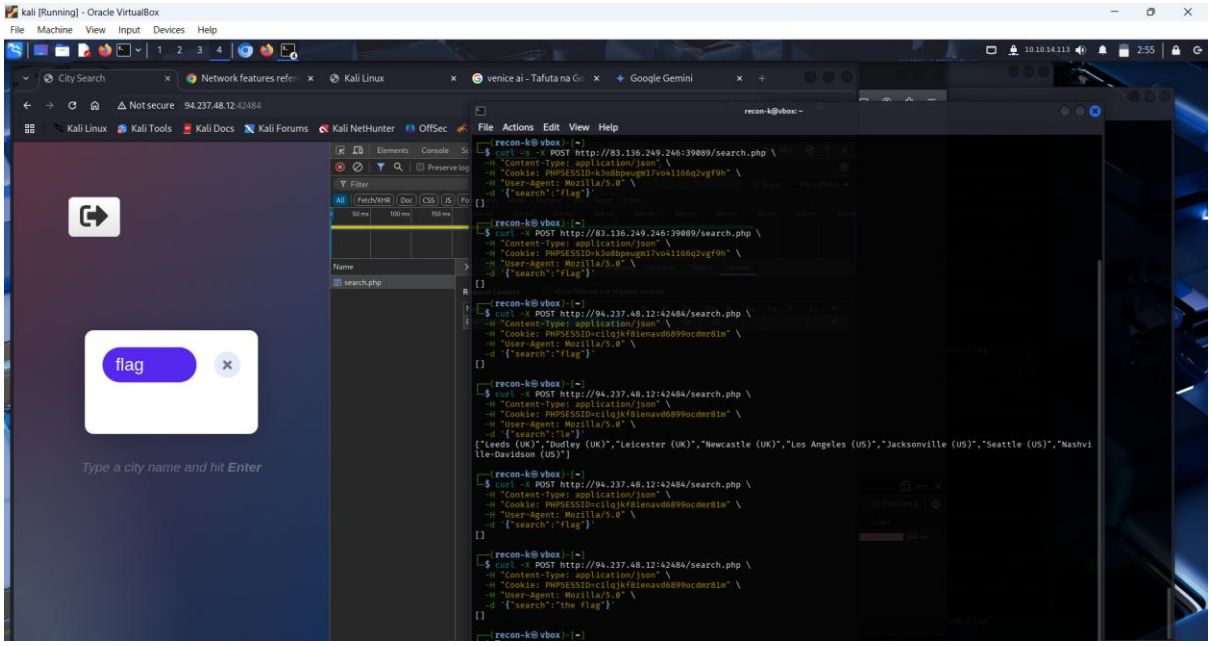
POST

Unlike HTTP GET, which places user parameters within the URL, HTTP POST places user parameters within the HTTP Request body



Questions

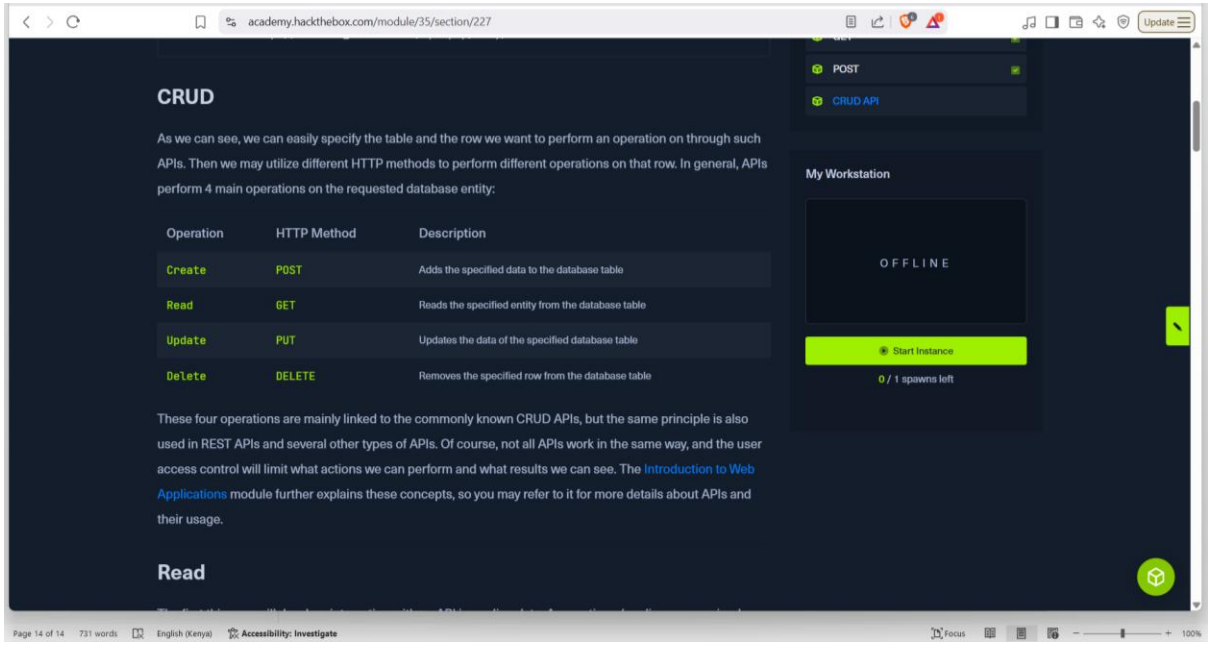
Obtain a session cookie through a valid login, and then use the cookie with cURL to search for the flag through a JSON POST request to '/search.php



CRUD API

CRUD stands for the 4 main operations done by APIs

Create, Read, Update, and Delete



Create commands

Update command

curl -X PUT http://<SERVER_IP>:<PORT>/api.php/city/london -d '{"city_name":"New_HTB_City", "country_name":"HTB"}' -H 'Content-Type: application/json'

Delete Command

curl -X DELETE http://<SERVER_IP>:<PORT>/api.php/city/New_HTB_City

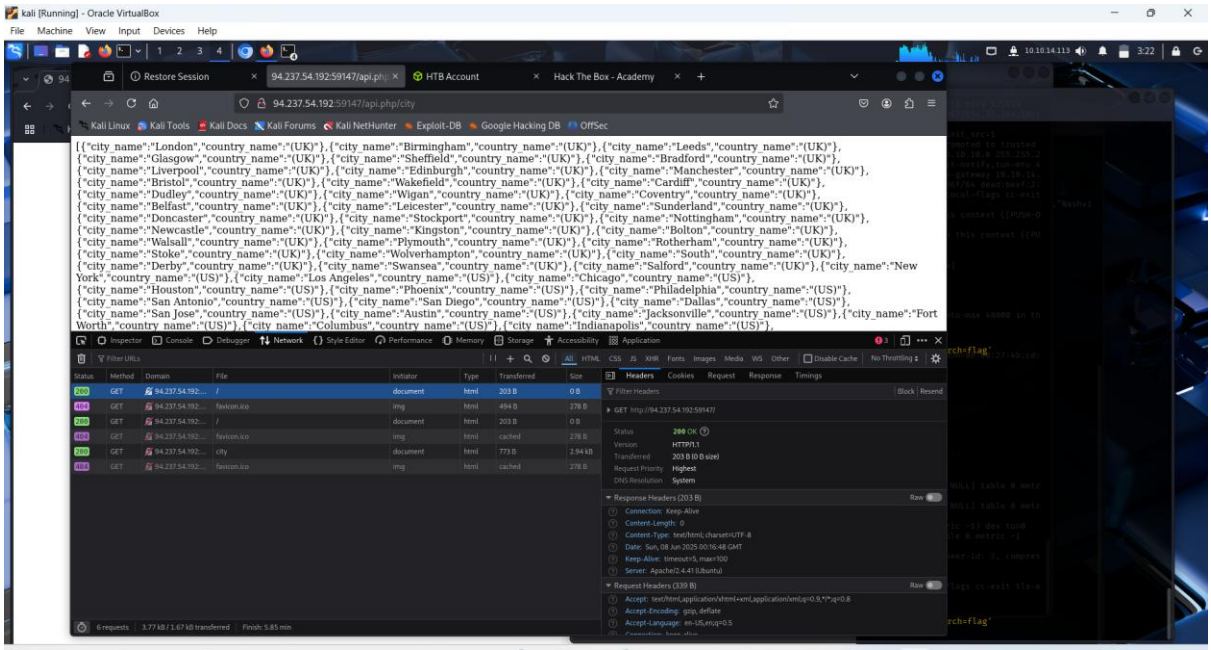
Questions

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.

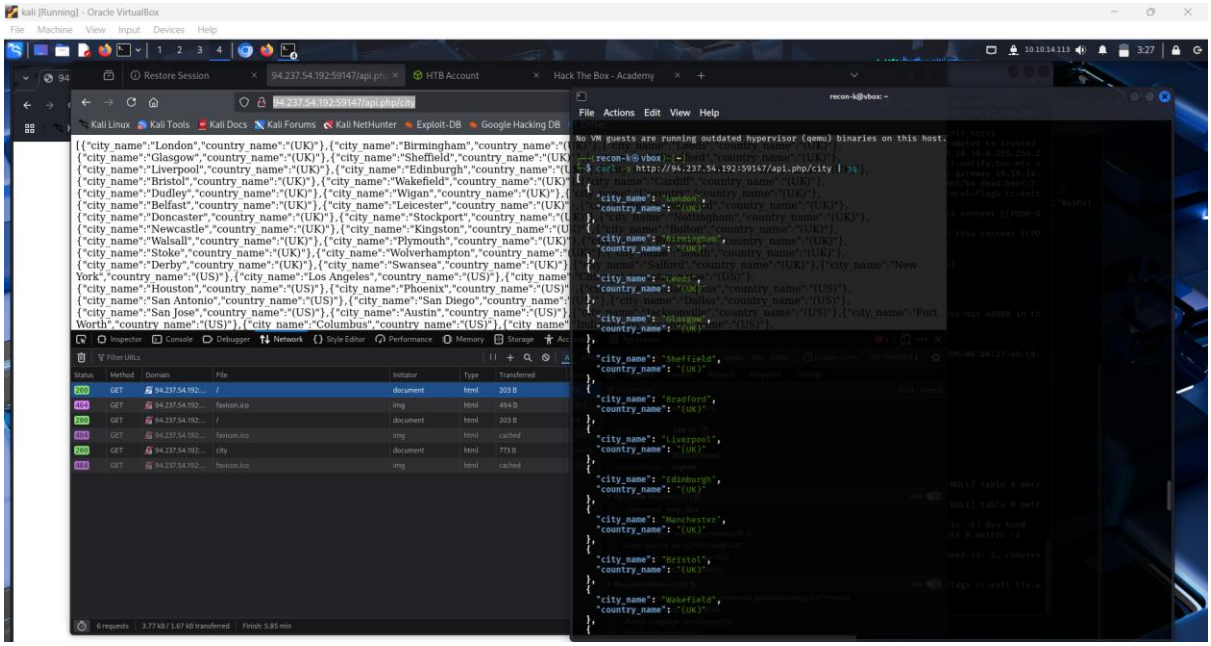
HTB{crud_4p!_m4n!pul4t0r}

Practical:

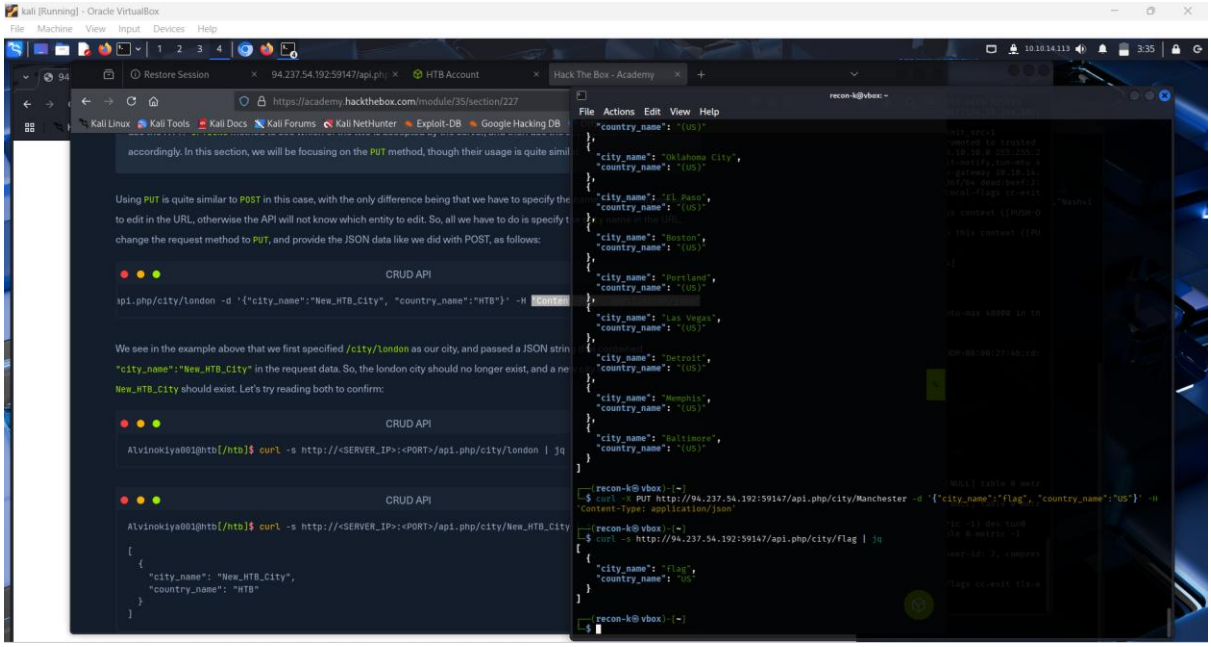
To solve the question, I started by opening the endpoint in a browser, and viewing the cities



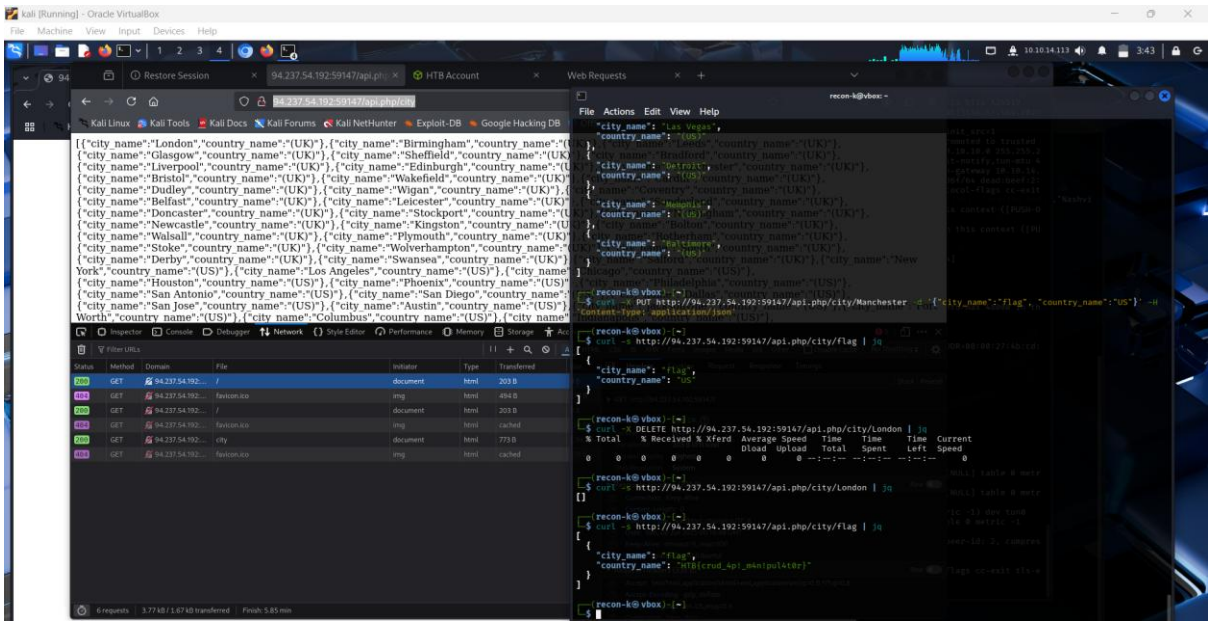
Then using the curl command with jq



I then updated the Manchester city name as flag



After that i Deleted London using the DELETE method and verified whether it exists to get my flag



Conclusion

This assignment provided valuable hands-on experience in understanding how web requests function and how they can be inspected and manipulated for testing purposes. I learned how different HTTP methods work, how headers and status codes communicate important information, and how to use tools like curl and browser dev tools effectively. These skills are foundational for web application security testing and will serve as a critical component in performing future assessments, especially when identifying vulnerabilities and analyzing web server behaviors.