Course: Cloud and Network Security - C3 – 2025

Student Name: Johnmark Gichuki

Student No: CS-CNS10-25101

Monday, September 29, 2025

Week 3 Assignment 2:

Class Exercise: HTB Academy - Web Requests

Table of Contents

Introduction

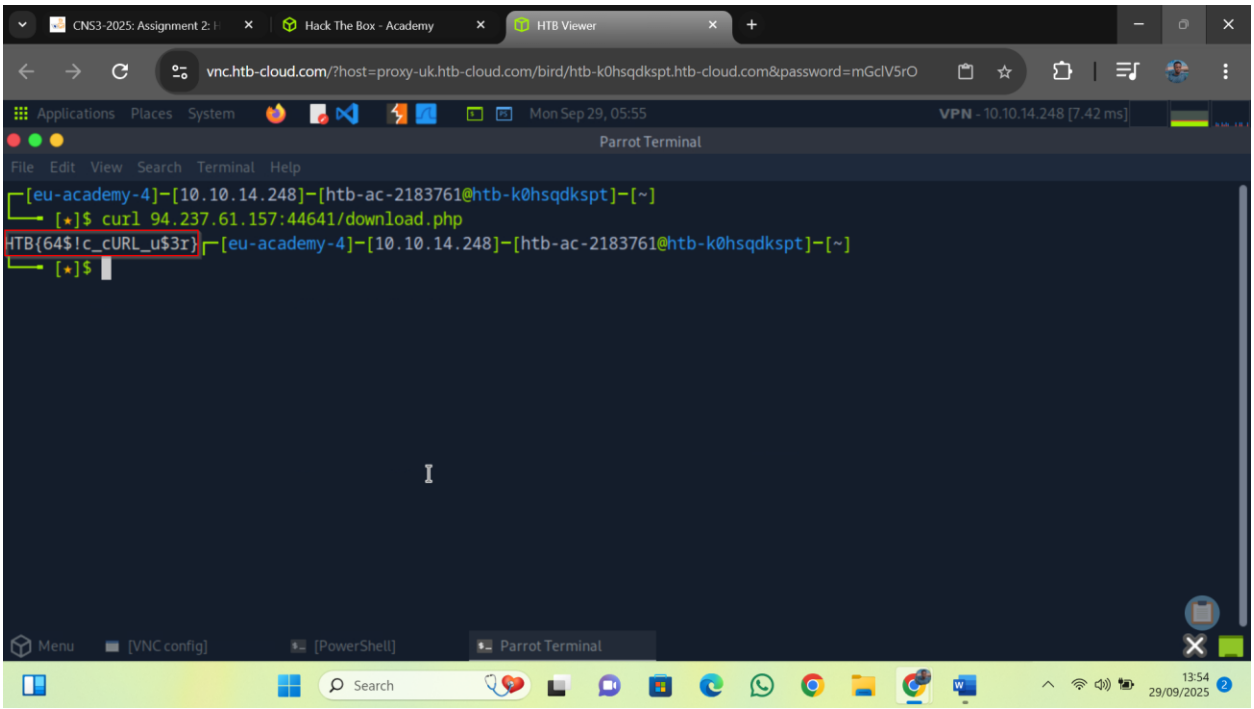
This assignment explores the foundational concepts and practical skills needed for web application security testing. The module focuses on how web applications communicate over HTTP and HTTPS, how requests and responses are structured, and how common HTTP methods (GET, POST, PUT, DELETE) and status codes behave. Rather than remaining purely theoretical, the coursework uses two core tools cURL and the Browser DevTools so you learn both how browsers interact with servers and how to reproduce and automate those interactions from the command line.

Hands-on exercises walk you through reading and crafting requests, inspecting headers, working with cookies and sessions, and interacting with simple APIs. The content is organised so you can practice each concept in isolation and then apply them together in end-of-module tasks and a skills assessment. Completing those exercises not only reinforces the mechanics of HTTP/HTTPS, it also builds the practical muscle memory required before attempting any real-world web application testing or vulnerability discovery.

Part 1: HyperText Transfer Protocol (HTTP)

Question

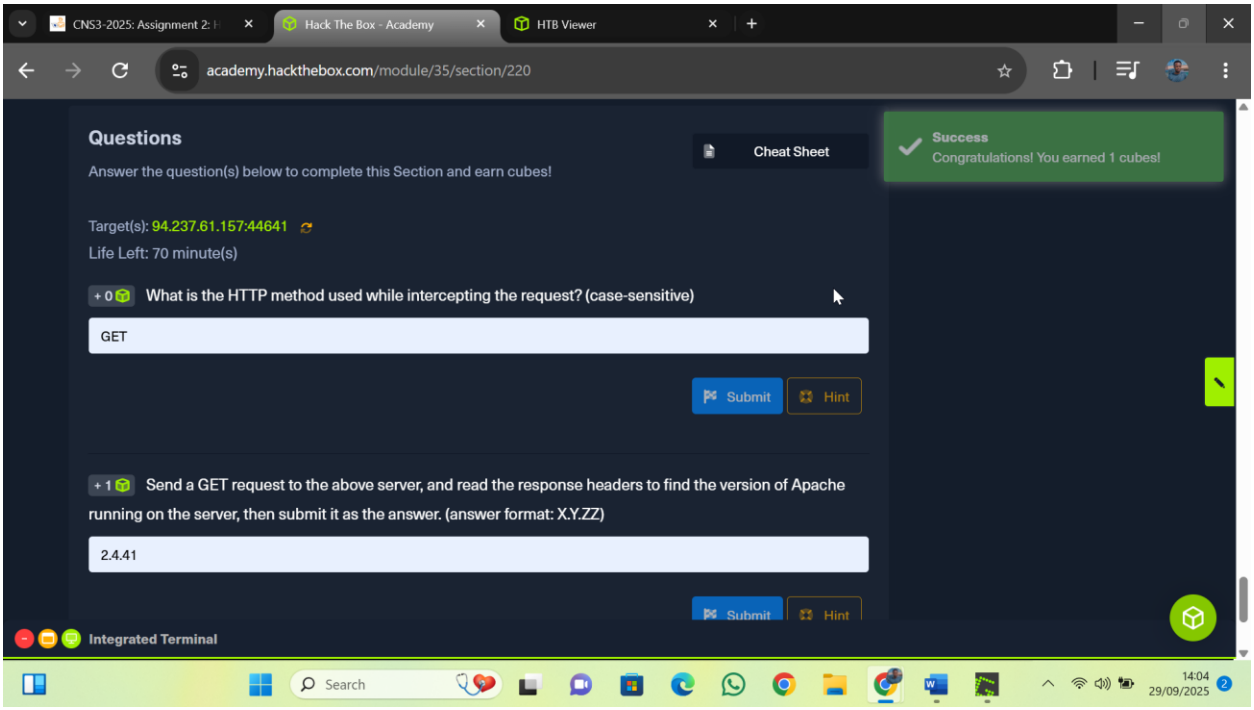
1. To get the flag, start the above exercise, then use cURL to download the file returned by '/download.php' in the server shown above. HTB{64$!c_cURL_u$3r}



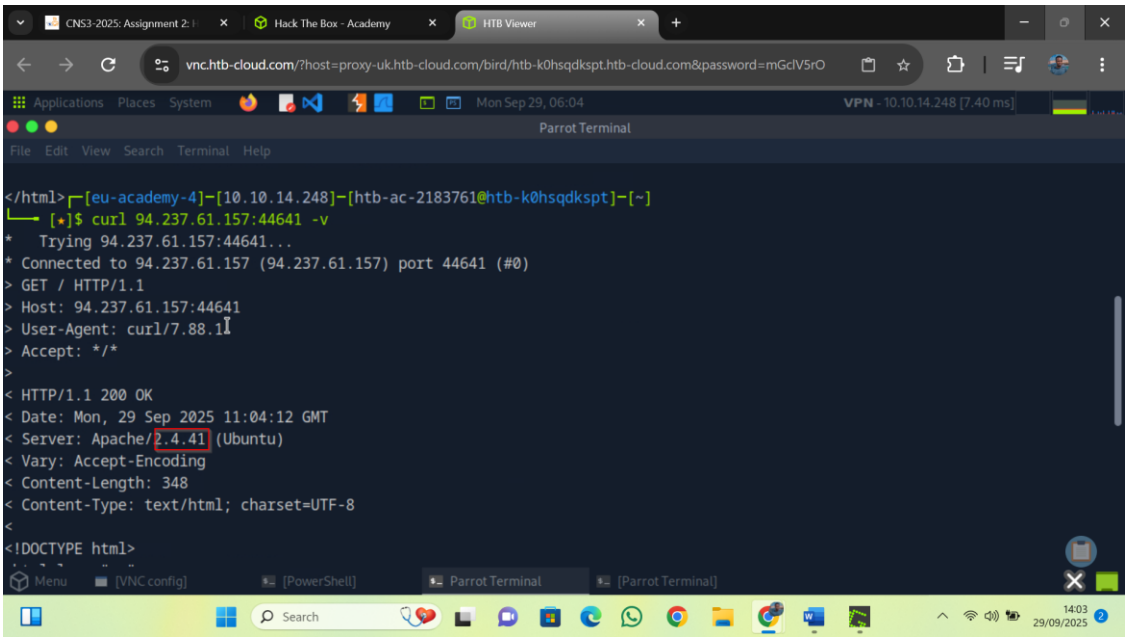
Part 2: HTTP Requests and Responses

Questions

1. What is the HTTP method used while intercepting the request? (case-sensitive) GET



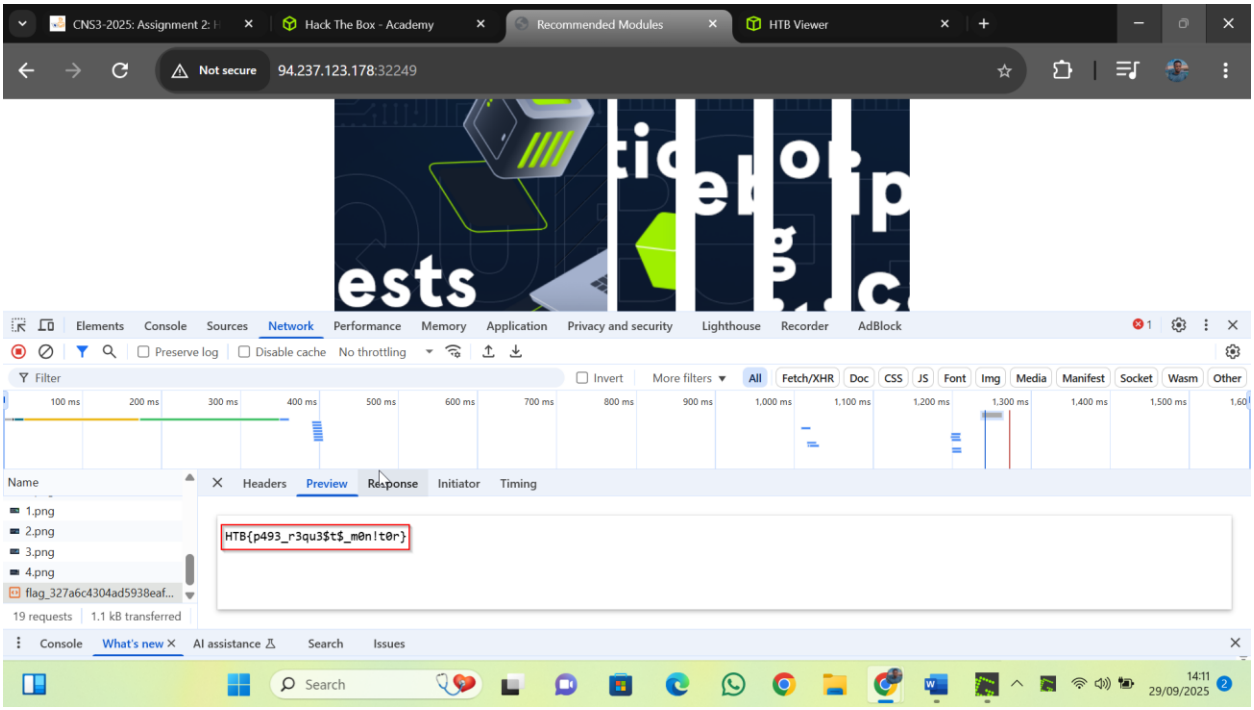
2. Send a GET request to the above server, and read the response headers to find the version of Apache running on the server, then submit it as the answer. (answer format: X.Y.ZZ) 2.4.41



Part 3: HTTP Headers

Questions

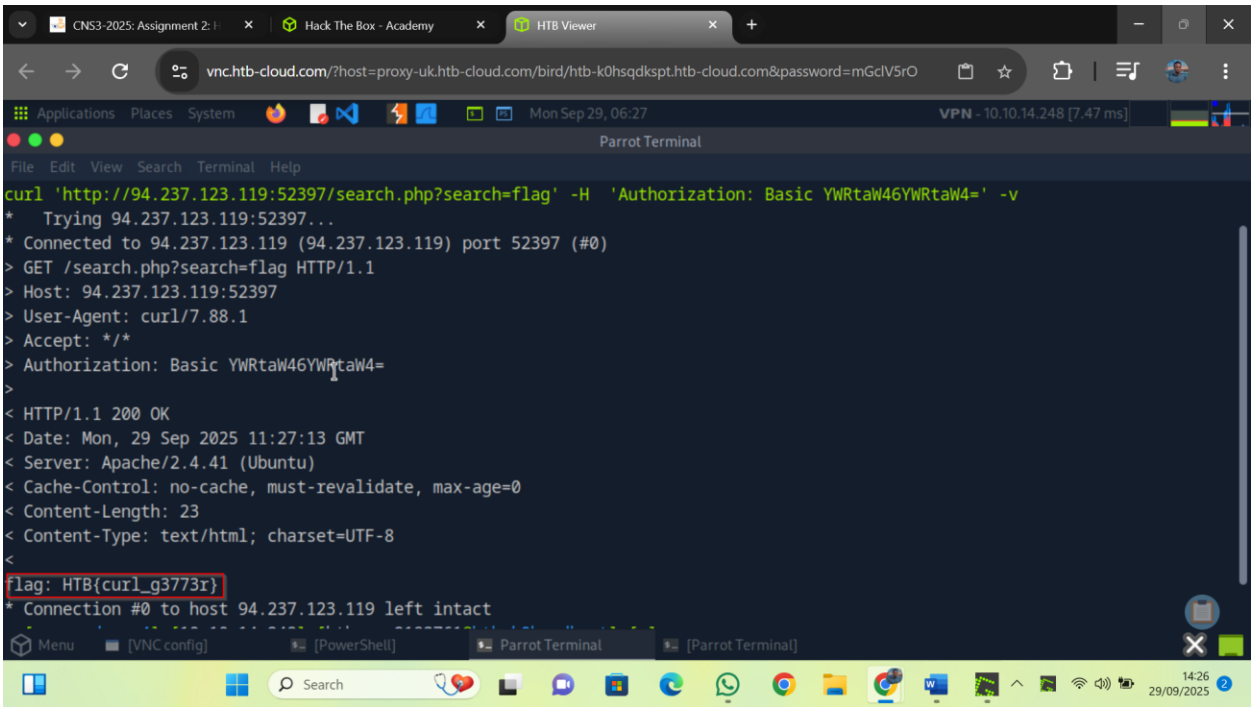
1. The server above loads the flag after the page is loaded. Use the Network tab in the browser devtools to see what requests are made by the page, and find the request to the flag. HTB{p493_r3qu3t_m0n!t0r}



Part 4: GET

Question

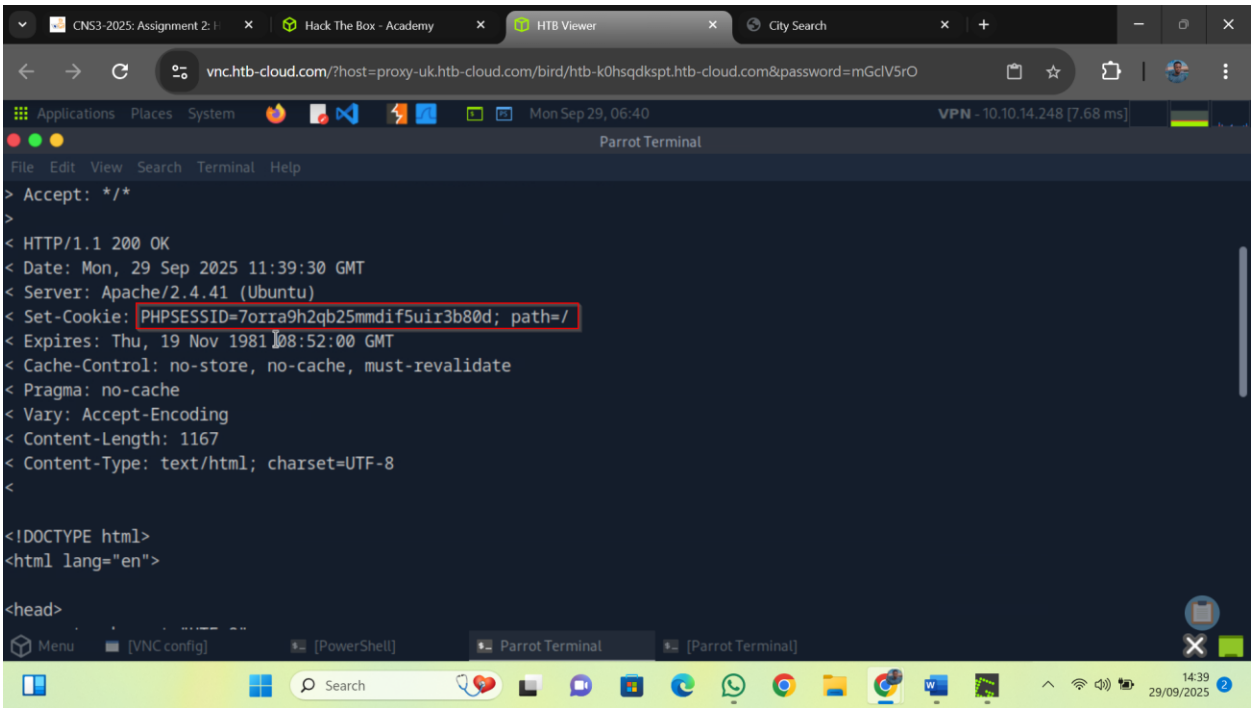
1. The exercise above seems to be broken, as it returns incorrect results. Use the browser devtools to see what is the request it is sending when we search, and use cURL to search for 'flag' and obtain the flag. HTB{curl_g3773r}



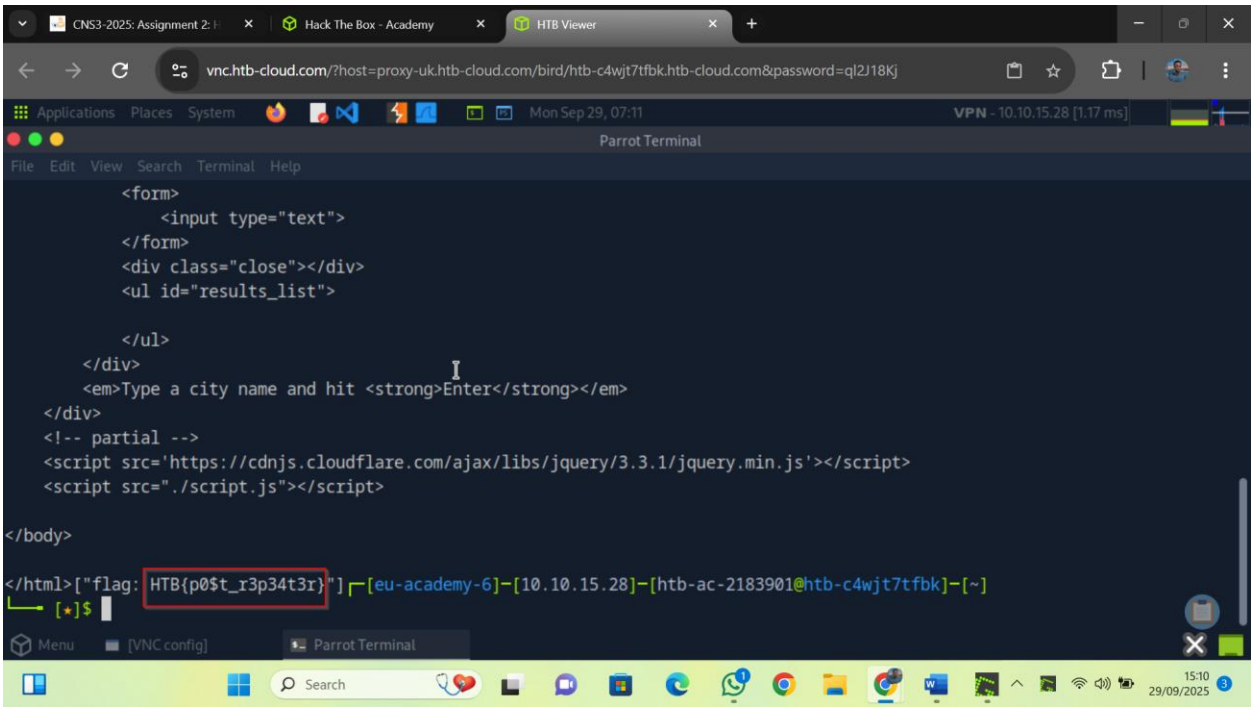
Part 5: POST

Questions

1. 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' HTB{p0$t_r3p34t3r}



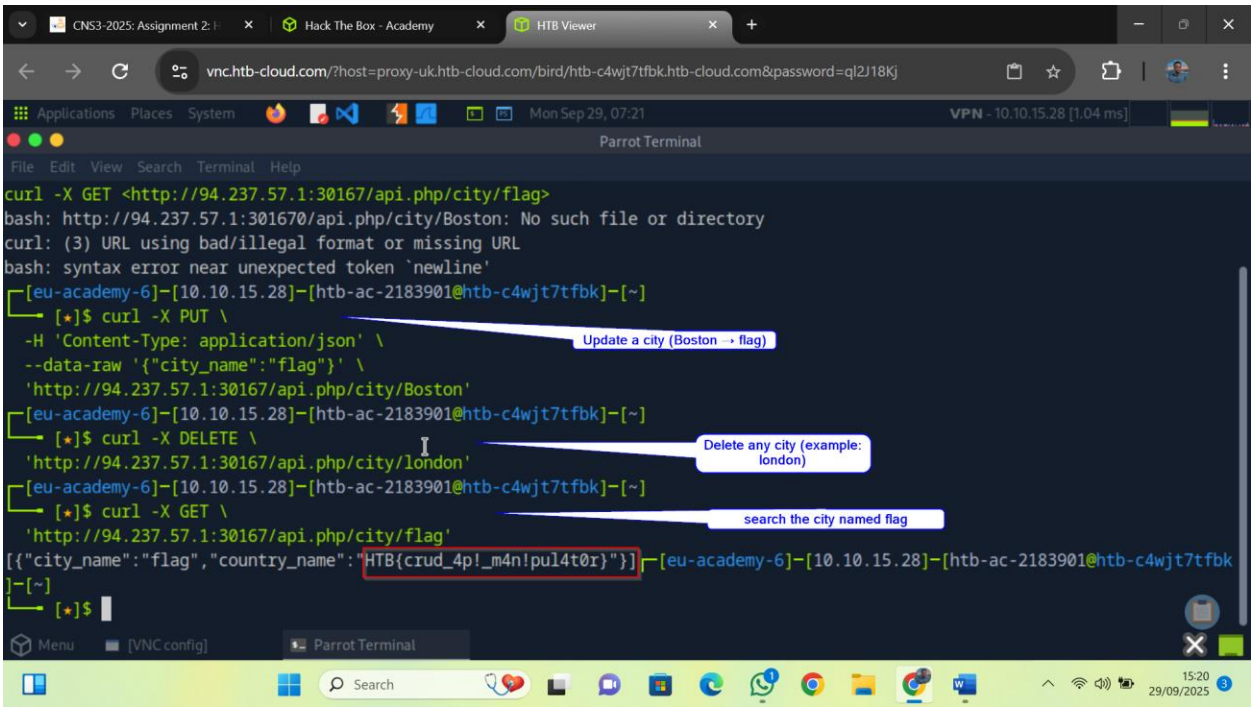
Once we have the cookie, we can use it in the subsequent request to search for the flag.



Part 6: CRUD API

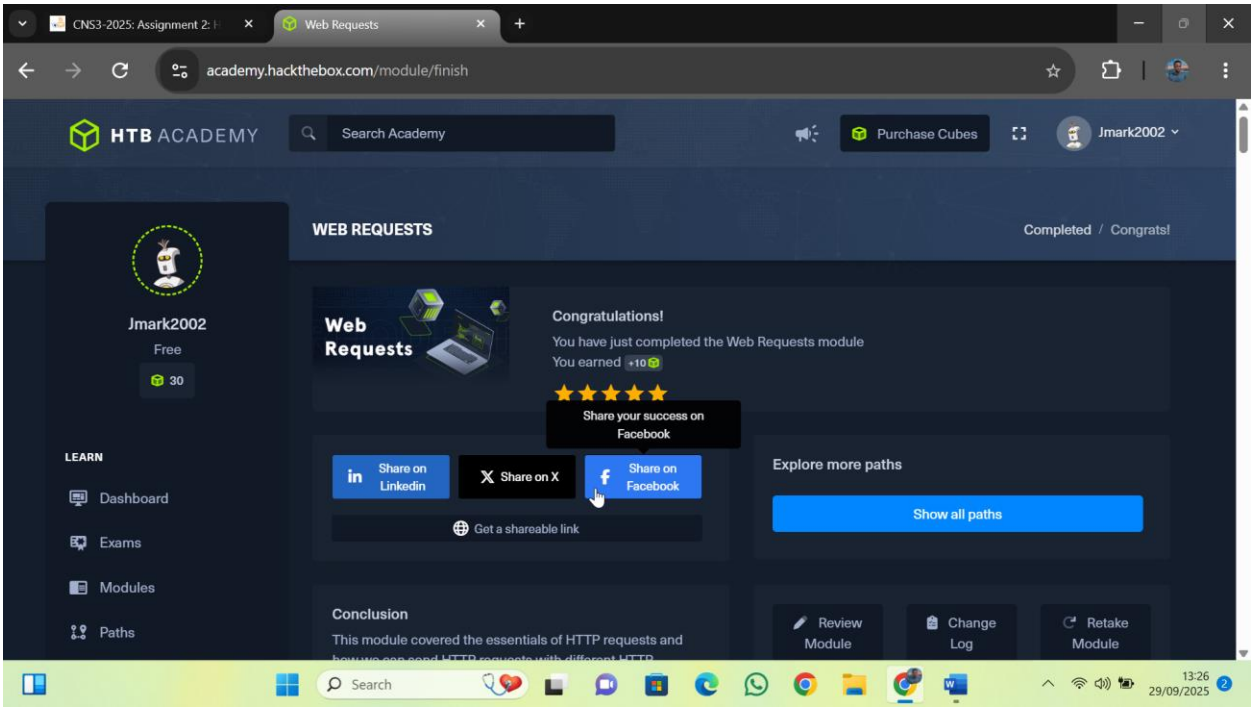
Questions

1. 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}



Completion Link and Image

Completion link: https://academy.hackthebox.com/achievement/2178420/35



Conclusion

Working through this module reinforced that effective web application testing begins with a solid understanding of how clients and servers communicate. I gained practical experience using Browser DevTools to observe real browser traffic and using cURL to replicate and automate those requests. That combination clarified how authentication (cookies/sessions), content types (like JSON), and HTTP verbs influence application behavior knowledge that is essential when probing for weaknesses or verifying fixes.

The hands-on exercises were especially valuable: they turned abstract protocol concepts into repeatable steps (capture request = modify/replicate = observe response) and helped me become comfortable with common troubleshooting techniques such as following redirects, viewing raw headers, and saving/using cookie jars. Overall, the module has improved my confidence with the tools and methods needed for basic web penetration testing and provided a clear foundation for more advanced topics like session management attacks, API testing, and automated scanning. Moving forward, I plan to practice these workflows regularly and apply them to increasingly complex targets to deepen my skills.