Internship Assignment Report: Cyber Security and Digital Forensics

Assignment 4: Burp Suite

- **TryHackMe Rooms:**
  - Burp Suite Repeater
- **Portswigger:**
  - lab-basic-te-cl

Assignment Overview

- Welcome to my internship assignment report! This document highlights the cybersecurity and digital forensics challenges I've tackled as part of my Cyber Security and Digital Forensics internship with **Cyber Secured India**. The focus of this report is on Burp Suite: Repeater, and it includes practical exercises completed on **TryHackMe**.
- Through these assignments, I've had the opportunity to apply theoretical knowledge in real-world scenarios, hone my problem-solving skills, and gain hands-on experience with essential cybersecurity concepts. Each section will walk you through the tasks I've completed, showcasing the steps I took, the tools I used, and the insights I gained along the way.

Background and Prior Experience

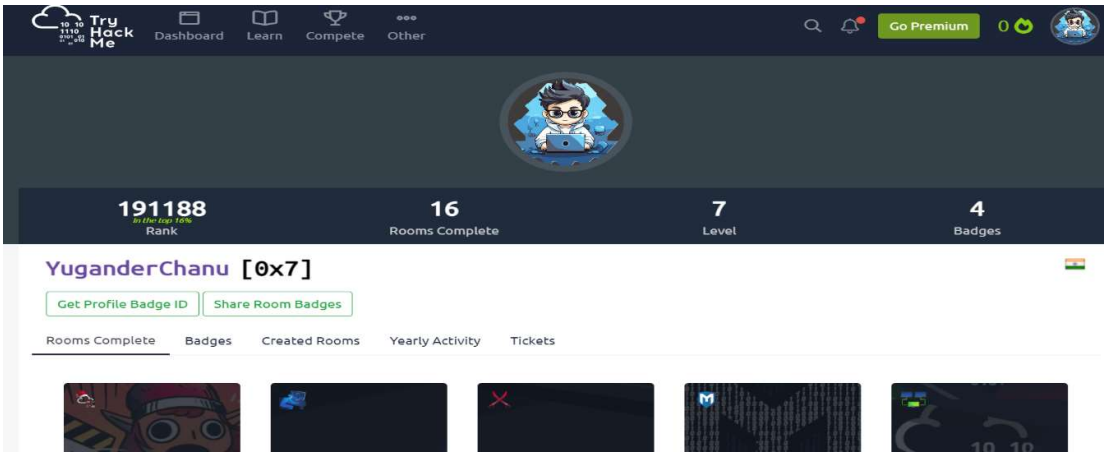
Before starting this internship, I had already completed several rooms on TryHackMe and HackTheBox. This prior experience gave me a solid foundation in cybersecurity principles and practical skills. During the internship, I continued to build on this knowledge by tackling new challenges and applying what I had previously learned in more complex scenarios.

About Me

- **Name:** Yugander Chanupalli
- **Position:** Cyber Security and Digital Forensics
- **Organization:** CyberSecured India
- **Email:** yugander9010@gmail.com
- **Submission Date:** 14/09/2024

TryHackMe

TryHackMe Account Picture:



Burp Suite: Repeater

Overview:

In the "Burp Suite Repeater" room on TryHackMe, I gained practical experience using Burp Suite's Repeater tool to manually modify and resend HTTP requests. This room focused on how Repeater helps in analyzing and altering requests to find potential vulnerabilities in web applications. The flexibility of the Repeater tool makes it an essential feature for penetration testers, as it allows for precise manipulation of web traffic, helping identify weaknesses in input validation, authentication mechanisms, and session management.

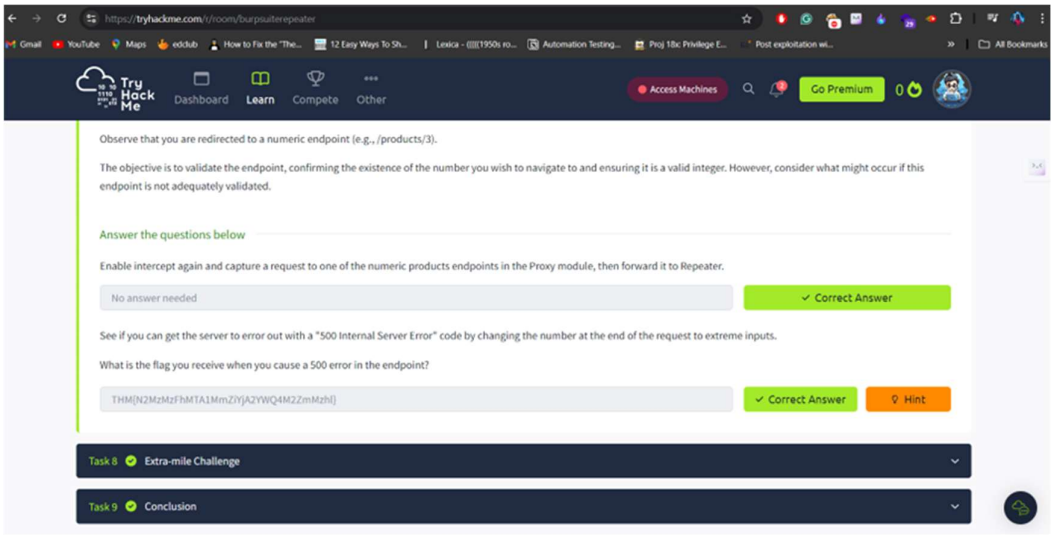
Key Techniques and Tools Learned:

Throughout this room, I focused on learning how to effectively use Burp Suite's Repeater tool, along with key techniques for manipulating HTTP requests:

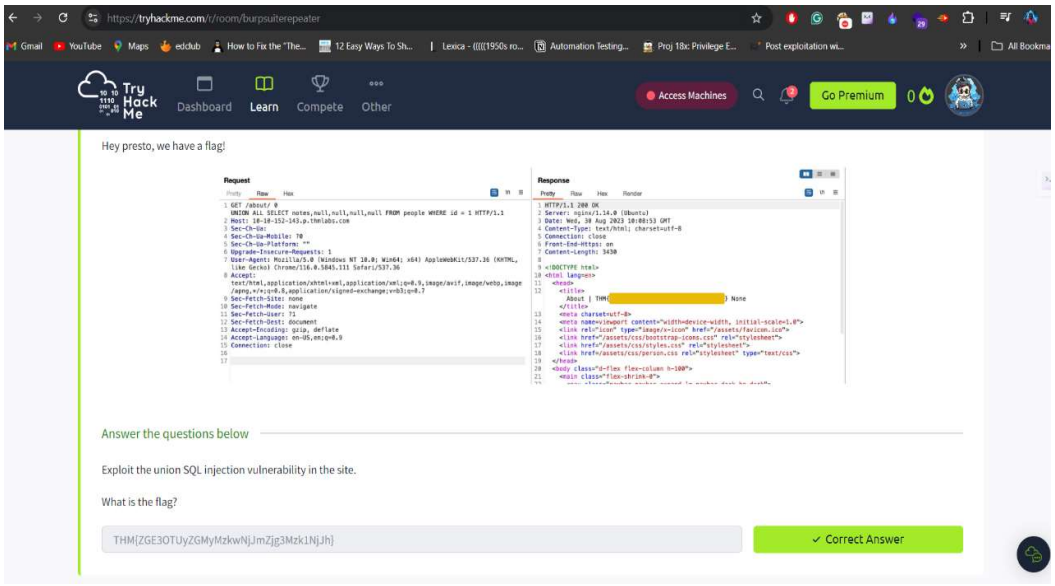
- **RepeaterTool:**
  Burp Suite's Repeater allows users to modify requests manually and resend them to the server to see how it responds. This is particularly useful for testing vulnerabilities like SQL injection, Cross-Site Scripting (XSS), and broken authentication mechanisms.
- **ModifyingRequests:**
  I learned how to change HTTP methods (GET, POST), alter headers, and manipulate parameters to test for different types of vulnerabilities in web applications. By tweaking these values, I was able to observe how different inputs affect the behavior of the server.
- **MessageAnalysis:**
  The message analysis feature allows for in-depth inspection of HTTP requests and responses, which is crucial for understanding how the server processes input. The ability to view headers, body, and even the raw data helped me spot potential vulnerabilities.
- **InspectorPane:**
  The Inspector pane made it easy to break down and modify different components of requests, such as parameters, cookies, and encoding. This helped me understand how these elements impact web application behavior and how they can be leveraged during penetration testing.

Proof of concept:

By solving the room I understood how burpsuite is important when it comes to interception of the requests. Below images allows you to understand my work.



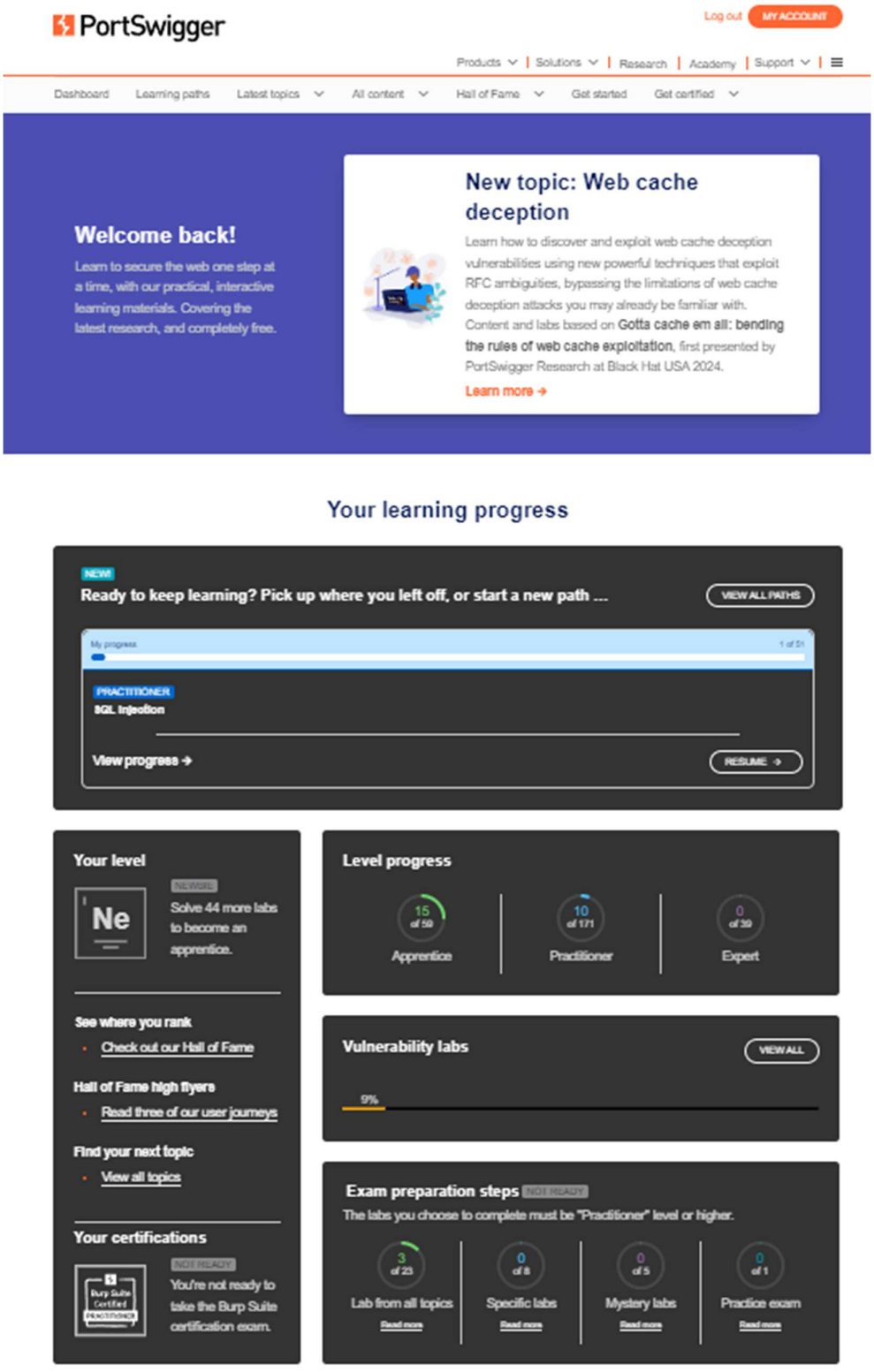
In this task, I manipulated the HTTP request using Burp Suite's Repeater tool by altering the request headers and parameters. This allowed me to observe the server's response and identify potential vulnerabilities in how the application handled input. After sending the modified request, I successfully captured a flag by exploiting weak input validation mechanisms.



During this challenge, I exploited a SQL injection vulnerability by injecting SQL queries into the input fields. By manipulating the request parameters through Burp Suite Repeater, I was able to bypass authentication and retrieve sensitive data from the database, which led to discovering the flag. This example demonstrated how improper validation of user input can lead to critical security breaches.

PortSwigger :

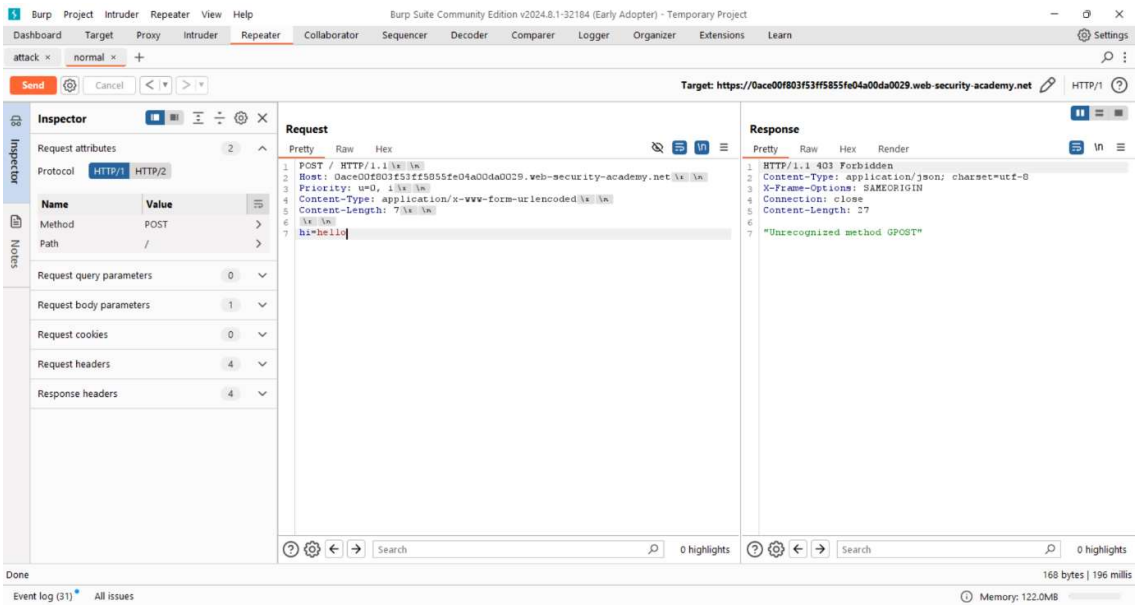
Before this room, I was already familiar with PortSwigger Labs, and I had solved labs on server-side vulnerabilities as well as SQL injection. The images below show my current progress.


PortSwigger
Log out
MY ACCOUNT
Products | Solutions | Research | Academy | Support
Dashboard
Learning paths
Latest topics
All content
Hall of Fame
Get started
Get certified
Welcome back!
Learn to secure the web one step at a time, with our practical, interactive learning materials. Covering the latest research, and completely free.
New topic: Web cache deception
Learn how to discover and exploit web cache deception vulnerabilities using new powerful techniques that exploit RFC ambiguities, bypassing the limitations of web cache deception attacks you may already be familiar with. Content and labs based on Gotta cache em all: bending the rules of web cache exploitation, first presented by PortSwigger Research at Black Hat USA 2024.
Learn more →
Your learning progress
NEW!
Ready to keep learning? Pick up where you left off, or start a new path ...
VIEW ALL PATHS
PRACTITIONER
SQL injection
View progress →
RESUME →
Your level
Ne
NEWBIE
Solve 44 more labs to become an apprentice.
See where you rank
Check out our Hall of Fame
Hall of Fame high flyers
Read three of our user journeys
Find your next topic
View all topics
Your certifications
NOT READY
You're not ready to take the Burp Suite certification exam.
Level progress
15 of 59
Apprentice
10 of 171
Practitioner
0 of 39
Expert
Vulnerability labs
VIEW ALL
9%
Exam preparation steps
NOT READY
The labs you choose to complete must be "Practitioner" level or higher.
3 of 23
Lab from all topics
Read more
0 of 8
Specific labs
Read more
0 of 5
Mystery labs
Read more
0 of 1
Practice exam
Read more

HTTP Request Smuggling (TE.CL):

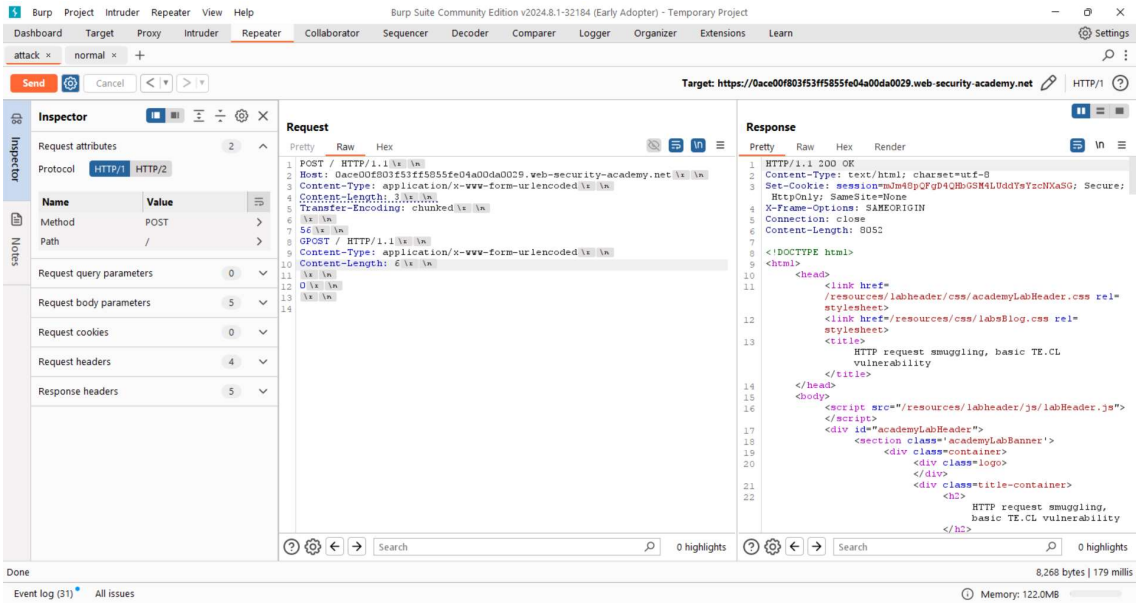
1. **Two Conflicting Headers**: The attacker uses both the Content-Length (CL) and Transfer-Encoding: chunked (TE) headers in a single request, which leads to a conflict.
2. **Different Interpretations**:
    - The **Proxy** interprets the request based on Content-Length, processing only part of the request.
    - The **Server** follows the Transfer-Encoding: chunked header, reading the entire request, including the hidden malicious part.
3. **Smuggling the Hidden Request**: The attacker hides a second malicious request within the body, which the proxy ignores but the server processes, leading to potential exploitation.

Proof of concept:

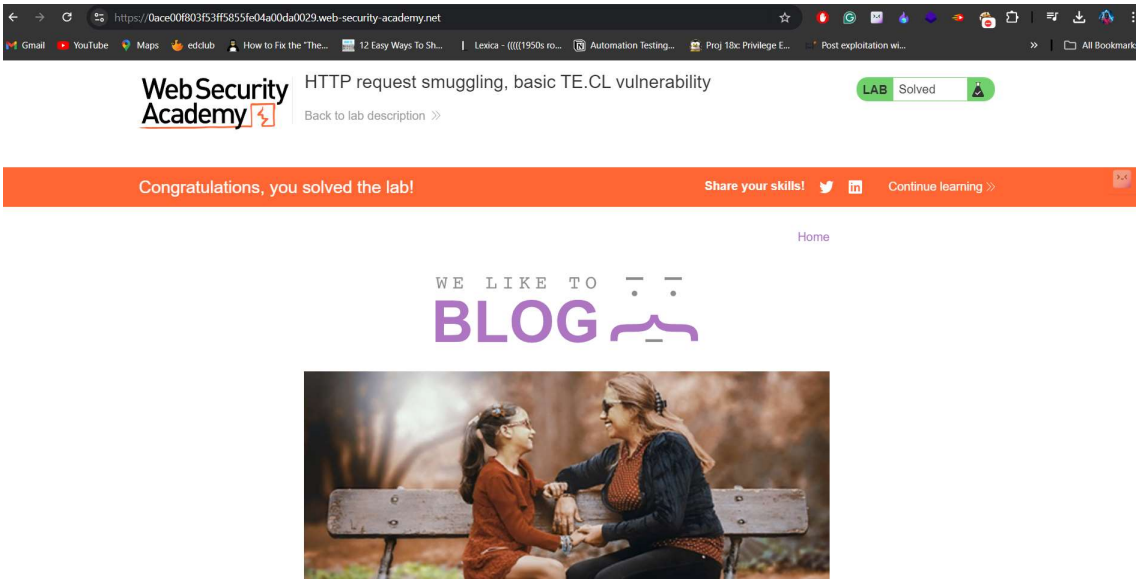


To complete this labs we have to make to requests on for attacking the server and another for reviewing wheather the attack is happened or not.

Normal Request : This request is crafted to test the server to understand the attack



Attack Request: It is responsible for the attack where I am sending the extra smuggled request to attack the server.



After the brief understanding about the attack I finally solved the lab.