

TryHackMe Walkthrough

Table of Contents

1. Overview of the challenge.....	3
▪ Introduction	3
2. Room Overview	4
3. Steps to exploit each vulnerability	6
▪ Task 1 - Bypass the Gate	6
➤ Steps to Capture the Flag	7
➤ Mitigation techniques.....	9
▪ Task 2 - A Message Left Behind	10
➤ Steps to Capture the Flag	11
➤ Mitigation techniques.....	13
▪ Task 3 - The Wrong File.....	14
➤ Steps to Capture the Flag	15
➤ Mitigation techniques.....	16
▪ Task 4 - Upload Surprise.....	17
➤ Steps to Capture the Flag	18
➤ Mitigation techniques.....	20
4. Explanation of flags.	21
▪ Flag Format	21
▪ Learning outcome.....	21

1. **Overview of the challenge.**

Room Link – <https://tryhackme.com/jr/lostinthm>

Room Name - LostInTHM — A Bugged Journey to Broke Web

Room Code - lostinthm

▪ **Introduction**

This report designs and the implementation of the CTF challenge room in TryHackMe. I have used a scenario-based incident to make it interesting. This challenge is tailored for FreshCart, an online grocery store. The main objective of the room is to create a learning environment to participants where they could investigate the incidents and uncover the flags.

The challenge was structured around four incidents, SQL injection, path traversal, Cross-site Scripting and insecure file upload. This report is focused only on web application vulnerabilities. These challenges were designed by aligning to OWASP Top 10, ensuring the learning areas are practical. For each task a flag was hidden, participants must analyze the scenario and exploit the vulnerability.

The site was developed using HTML, CSS, JS and PHP. After testing the site in the local machine using Apache it was then hosted in infinity free for easy access. Since it is a free service, site might not always be available therefore I have added another option in case the site doesn't work players can use the docker to access the site and do the tasks.


Finally, this report is structured with four vulnerabilities, outlining the steps of exploiting them and the process of capturing the flag.

2. Room Overview

Cyber awareness month deal: 5 months free on annual subscriptions

Subscribe Now

32 : 02 : 36
hr min sec

 TryHackMe

Dashboard

Learn

Practice

Compete

Develop

Access Machines

Go Premium

0

L

Learn > LostInTHM — A Bugged Journey to Broke Web

 **LostInTHM — A Bugged Journey to Broke Web**
A short, hands-on web-security room where you investigate a small Online Grocery Store app full of realistic bugs. Find and exploit four vulnerabilities to recover progressively harder flags and learn practical fixes.
📶 ⌚ 60 min 👤 2 🔒

Start AttackBox

Help

Save Room

Options

Room progress (0%)

Chart

Scoreboard

Write-ups



Task 1 ☐ Introduction

Task 2 ☐ Bypass the Gate

Task 3 ☐ A Message Left Behind

Task 4 ☐ The Wrong File

Task 5 ☐ Upload Surprise

TryHackMe Walkthrough

Task 1 Introduction

You are a security investigator contracted by FreshCart, a small online grocery delivery startup. FreshCart has reported unusual behaviour: an admin-only note has disappeared, a staff-only messages page may reveal secrets, and uploaded customer files are behaving oddly. Your job is to help the FreshCart team by investigating four incidents, each of which hides a token you must retrieve.

This challenge teaches practical exploitation techniques, demonstrates how vulnerabilities are chained in real-world scenarios, and highlights best practices for securing web applications. By the end of the room, you will have hands-on experience in discovering, exploiting, and understanding mitigations for critical web security issues mapped to the [OWASP Top 10](#).

Room goals

- Complete four tasks to retrieve flags.
- Learn practical exploitation techniques and, most importantly, the correct mitigations.

There are two options if the direct site doesn't work use the docker

Option - 1

Direct access to the site - <https://freecart.gt.tc/>

Option - 2

Download the zip file given and unzip it.

HOW TO RUN:

1. Install Docker Desktop from <https://docker.com>
2. Open **Docker Desktop**
3. Open the terminal from the freshCart folder.
4. Run: **docker-compose up -d**
5. Open: <http://localhost:8000>

TO STOP:

Run: **docker-compose down**

Answer the questions below

Overview Understood?

No answer needed

Complete

Players have two options, if the direct access site does not work players can use option two. When using option two first players must download the tasks files and unzip it. And then they must download and install docker desktop application from the given link. Open the terminal from the tasks folder and run **docker-compose up -d** to start the docker. Players can access the site from <http://localhost:8000> . To stop the docker players can run **docker-compose down** in the terminal.

3. Steps to exploit each vulnerability.

▪ Task 1 - Bypass the Gate

Objective - Gain access to a restricted area(profile) and retrieve the hidden flag.

Hints - Think about how input might be combined to form a query.

Details

SQL injection is a flaw included in SQL queries without proper sanitization. Login bypass is a common SQLi use where authentication is manipulated so it returns a valid user. Without checking the provided login credentials, the payload causes the WHERE clause to evaluate as true, which gives access to the user profile. When the player bypasses the login, flag reveals.

Overview

Task 2 ○ Bypass the Gate ^

The FreshCart manager locked down the admin dashboard after a suspicious login attempt. They think the door is misconfigured and want you to check whether authentication can be bypassed.

If **option 1** is used

Navigate to - <https://freecart.gt.tc/pages/signin.php>

If **option 2** is used

Navigate to - <http://localhost:8000/index.html>

Try access a user account and retrieve the hidden flag.

Allowed actions: Submit the form repeatedly, try different inputs, observe resulting pages and redirects. Use the browser and standard tools (Browser DevTools, Burp) if you like.

Answer the questions below

submit the flag as "flag: THM {-----}"

Answer format: `***:***{***** _****}`

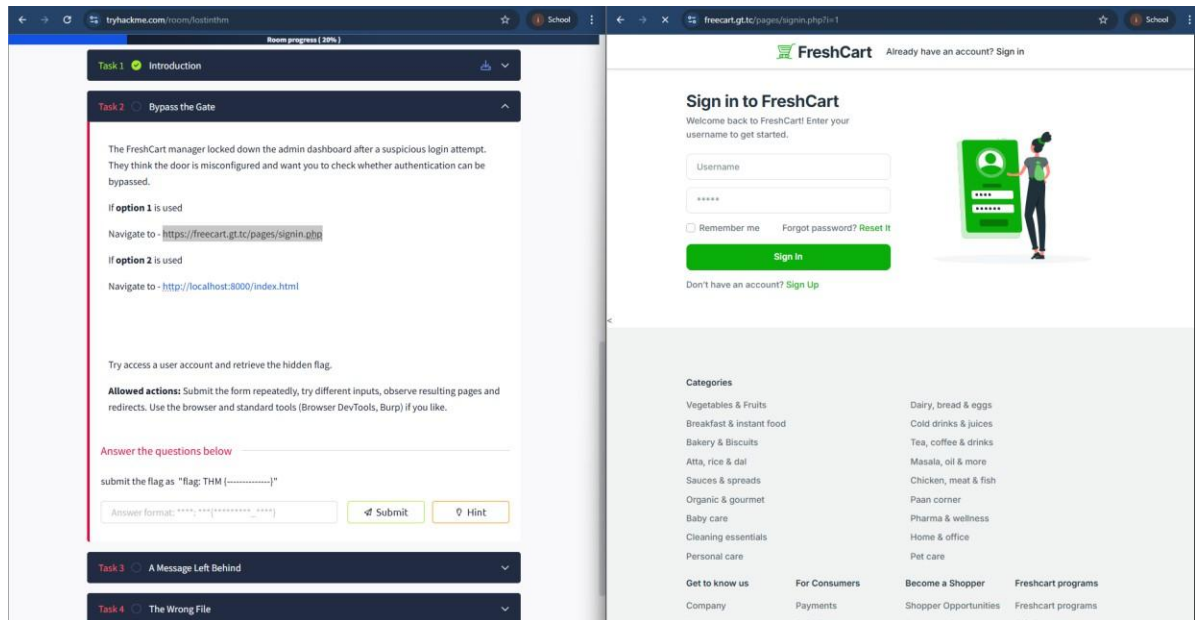
Submit

Hint

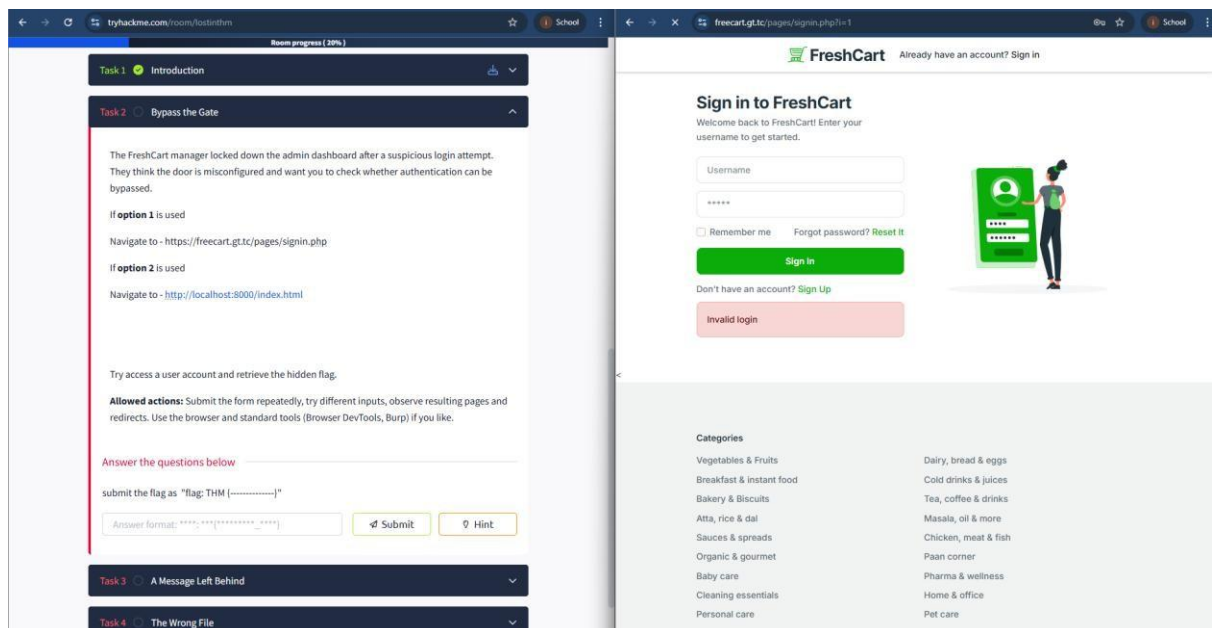
TryHackMe Walkthrough

➤ Steps to Capture the Flag

Step 1- Navigate to the Signin page through the link given.



Step 2 – Try different combinations for username and passwords.



TryHackMe Walkthrough

Step 3 - Try a simple payload. (' OR 1=1-- / ' OR '1'=1)

freecart.gt.tc/pages/signin.php

FreshCart Already have an account? Sign in

Sign in to FreshCart

Welcome back to FreshCart! Enter your username to get started.

' OR 1=1--

☐ Remember me Forgot password? [Reset It](#)

Sign In

Don't have an account? [Sign Up](#)

tryhackme.com/room/lostinthm

freecart.gt.tc/pages/account-orders.php?flag=THM%7BlostInTHM_sql%7D

Room progress (20%)

Task 1 Introduction

Task 2 Bypass the Gate

The FreshCart manager locked down the admin dashboard after a suspicious login attempt. They think the door is misconfigured and want you to check whether authentication can be bypassed.

If **option 1** is used

Navigate to - <https://freecart.gt.tc/pages/signin.php>

If **option 2** is used

Navigate to - <http://localhost:8000/index.html>

Try access a user account and retrieve the hidden flag.

Allowed actions: Submit the form repeatedly, try different inputs, observe resulting pages and redirects. Use the browser and standard tools (Browser DevTools, Burp) if you like.

Answer the questions below

submit the flag as "flag: THM {.....}"

Answer format: `****,****(***** ****)`

Submit **Hint**

Task 3 A Message Left Behind

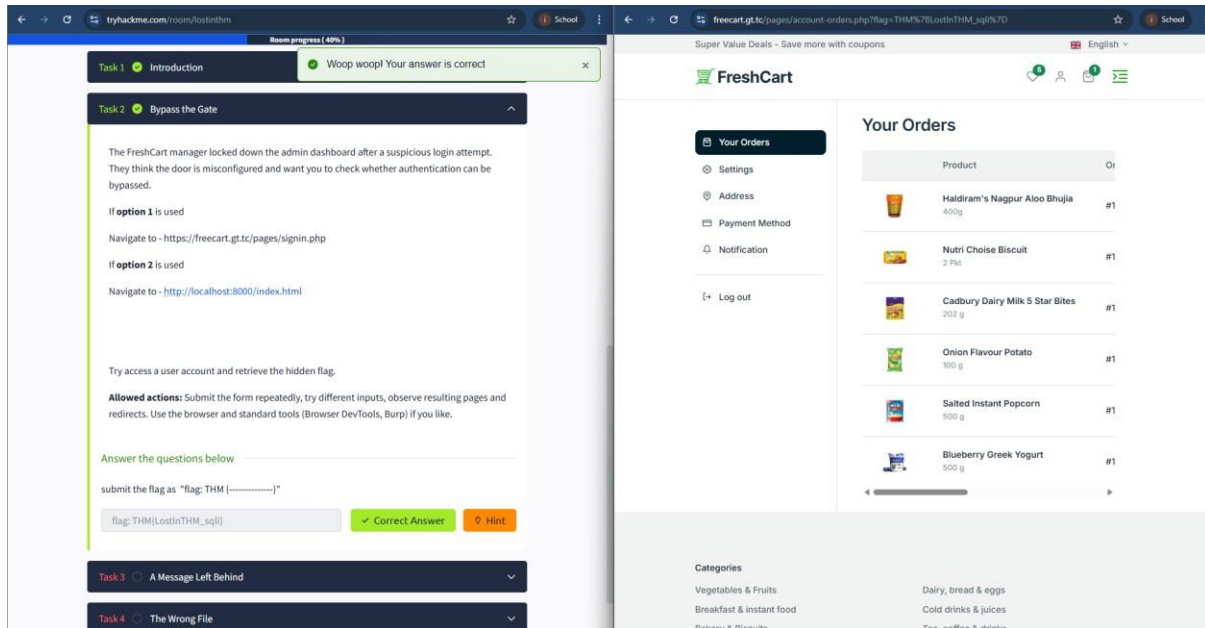
Task 4 The Wrong File

freecart.gt.tc says
Flag: THM(lostInTHM_sql)

OK

TryHackMe Walkthrough

Step 4 – Copy- paste the flag to submit the flag.



➤ Mitigation techniques

- To avoid string concatenation, use parameterized queries.
- Use input validation and least privilege principle to database.
- Enable a firewall to the web application to detect suspicious loggings.

▪ Task 2 - A Message Left Behind

Objective - Submit a message via the contact form that reveals the flag when a form validation is not done.

Hints – Try different characters and HTML-like input in message content.

Details

Cross-Site Scripting (XSS) occurs when an attacker injects a JavaScript payload to user fields and application renders it without proper encoding. When the user loads that page the payload loads with its content. Here the vulnerability is in the contact form, so in this challenge when the JavaScript is loaded the flag appears. Reflected XSS is used here so that right after the payload is sent it takes response.

Overview

Task 3 A Message Left Behind

A customer support note sent through the contact form accidentally contains a secret. Staff view stored messages in a staff-only page. The manager asks you to leave a message that will reveal a message when a staff member reads it.

if **option 1** is used..

Navigate to - <https://freecart.gt.tc/pages/contact.php>

if **option 2** is used..

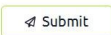

Navigate to - <http://localhost:8000/pages/contact.php>

Try to retrieve the hidden flag.

Answer the questions below

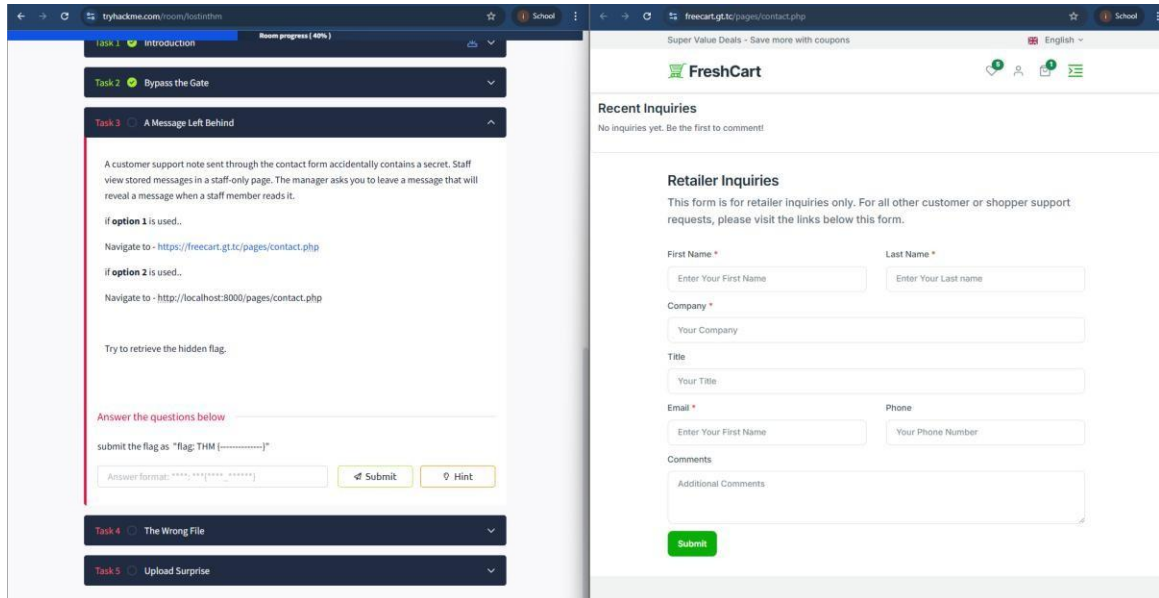
submit the flag as "flag: THM {-----}"

Answer format: ****.***[****_*****]

➤ Steps to Capture the Flag

Step 1- Navigate to the contact page.



Step 2 – Try sending a inquiry and monitor how it appears.

Super Value Deals - Save more with coupons

English

FreshCart

Recent Inquiries

Hi Jim

Retailer Inquiries

This form is for retailer inquiries only. For all other customer or shopper support requests, please visit the links below this form.

First Name *

Last Name *

Enter Your First Name

Enter Your Last name

Company *

Your Company

Title

Your Title

Email *

Phone

Enter Your First Name

Your Phone Number

Comments

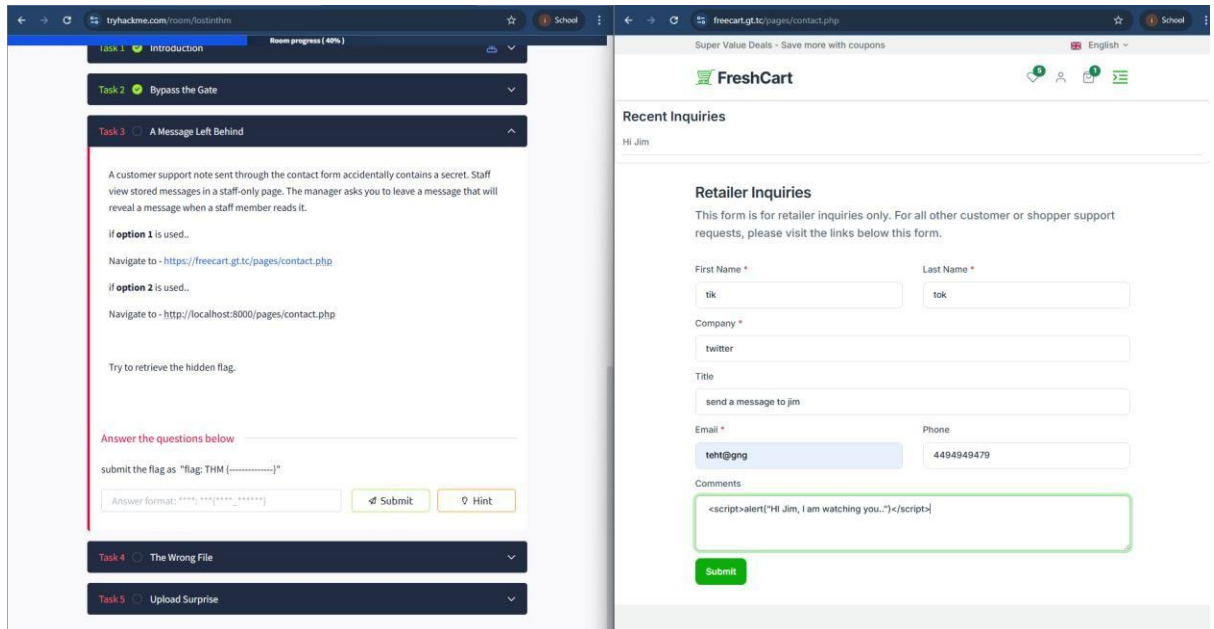
Additional Comments

Submit

TryHackMe Walkthrough

Step 3 - Now try submitting a simple inline JavaScript.

```
<script>alert("HI Jim, I am watching you..")</script>
```



Super Value Deals - Save more with coupons

English

FreshCart

Recent Inquiries

FLAG: THM{lost_XSS#14}

Retailer Inquiries

This form is for retailer inquiries only. For all other customer or shopper support requests, please visit the links below this form.

First Name *

Last Name *

Enter Your First Name

Enter Your Last name

Company *

Title

Your Company

Your Title

Email *

Phone

Enter Your First Name

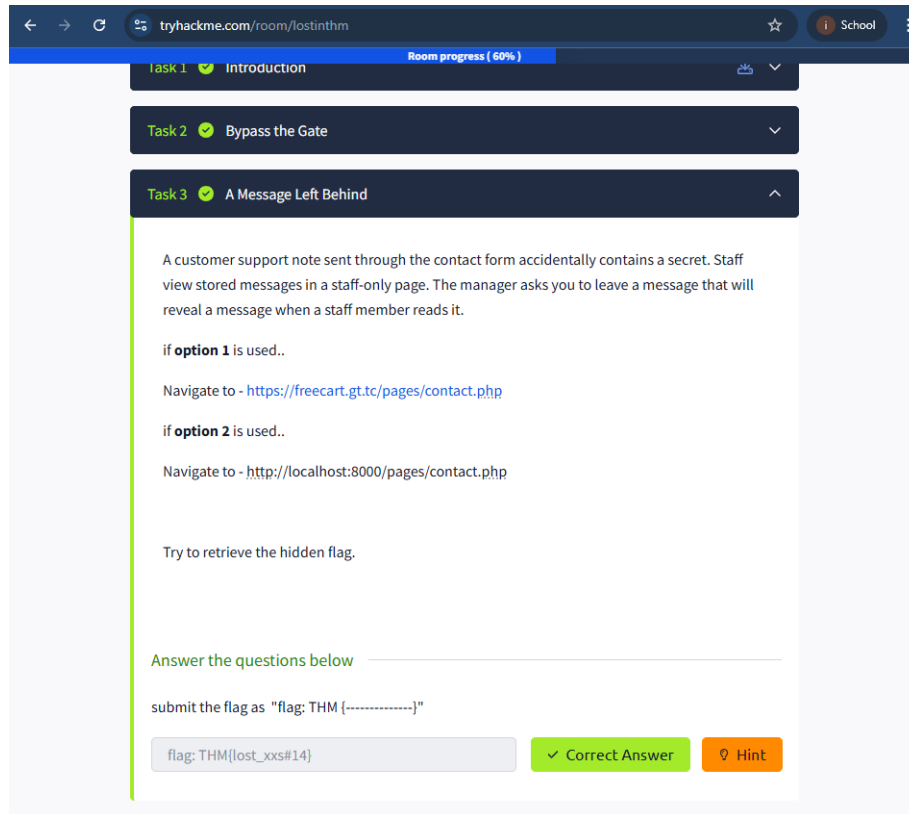
Your Phone Number

Comments

Additional Comments

Submit

Step 4 – Submit the flag to complete the task.



➤ Mitigation techniques

- Implement a strong input validation and sanitization.
- Enforce a Content Security Policy (CSP) header to limit the allowed scripts.
- Allow only HTTPS requests and secure cookies to avoid theft.
- Always encode the output to avoid injecting scripts.

▪ Task 3 - The Wrong File

Objective - Navigate through the download page and find the hidden flag.

Hints - Experiment with directory navigation patterns and encodings.

Details

Path traversal or directory traversal is when an endpoint concatenates a user supplied filename path without restricting it. An attacker may use ../ segment to move up in the directory tree to access files. Here the flag is hidden in a text file and the player has to search through the folders to find the flag and then submit it to complete the task.

Overview

Task 4 ○ The Wrong File ^

A delivery driver accidentally uploaded a private report to a public download endpoint. Files should be restricted, but the download endpoint appears to accept arbitrary file paths.

if **option 1** is used..

Navigate to - <https://freecart.gt.tc/download.php>

if **option 2** is used..

Navigate to - <http://localhost:8000/download.php>

Try to retrieve the hidden flag.

Answer the questions below

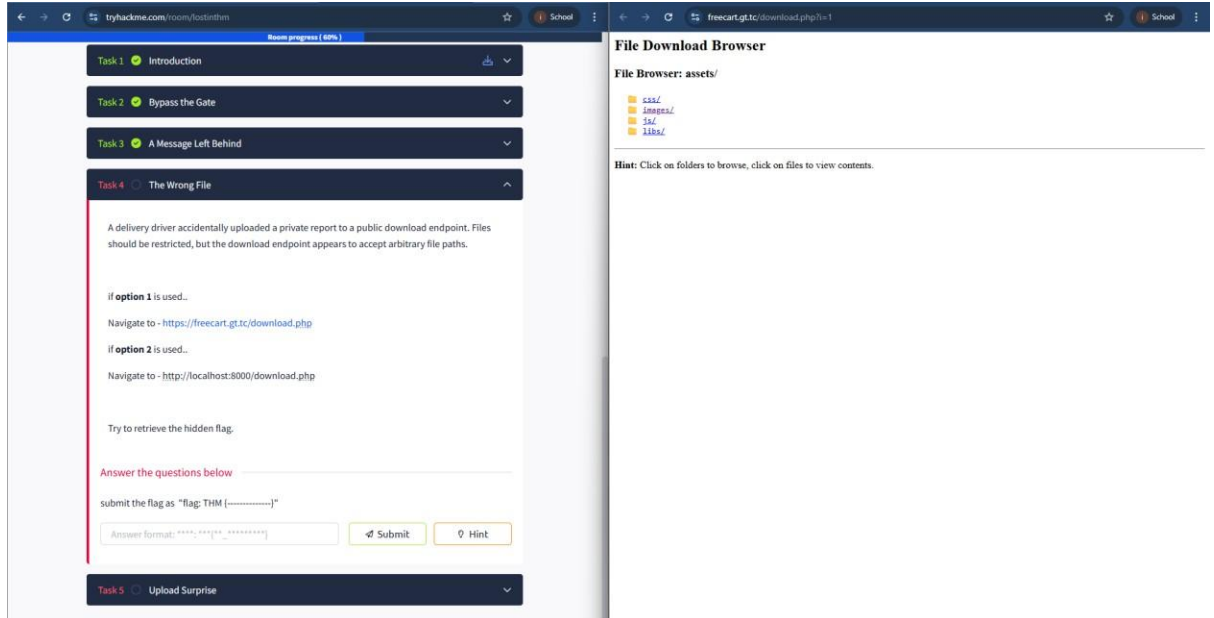
submit the flag as "flag: THM {-----}"

Answer format: ****.***{**_*****}

Submit Hint

➤ Steps to Capture the Flag

Step 1 – Navigate to the link given.



Step 2 – Search for any text files in the folders.



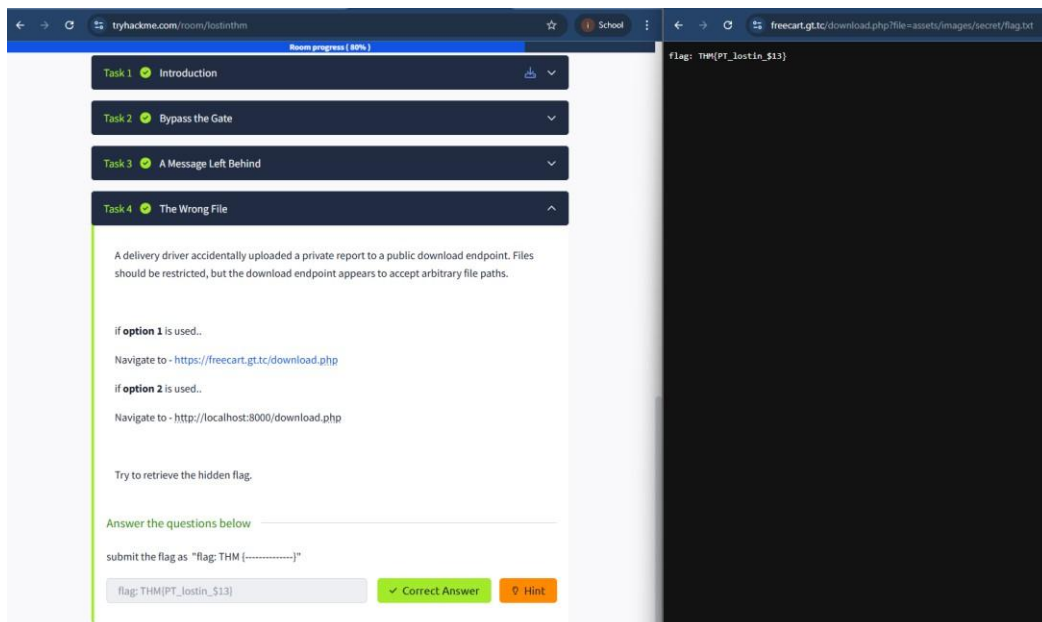
Here there is a suspicious folder called “Secret”. Check on that.

Step 3 - Navigate to the suspicious folder.



Here we have found the file containing the flag.

Step 4 – Open the file which contains the flag and submit it.



➤ Mitigation techniques

- Canonicalize the file paths in the server side to remove ../ segments.
- Permit access only to known filenames by implementing a allow list.
- Store important files separately from the root and restrict permissions.
- Enforce least privilege principle to restrict access to sensitive information and log the suspicious activities.

▪ Task 4 - Upload Surprise

Objective - Upload a file through the customer profile upload that allows you to retrieve the flag.

Hints - Try uploading text-based files first and then check their URL to download the file.

Details

Insecure file upload is a vulnerability caused due to poor validation in file type and content. A web shell file is a small script that provides the command execution through that relevant site. In this challenge attacker tries to gain access to the content which is hidden in the site through a php file.

Overview

Task 5 ☐ Upload Surprise

The FreshCart customer profile form accepts profile pictures. The operations team suspects uploaded files may be executed or serve sensitive content unintentionally. They ask you to deliver a file that reveals a token.

if **option 1** is used..

Navigate to - <https://freecart.gt.tc/dashboard/create-customers.html>

if **option 2** is used..

Navigate to - <http://localhost:8000/dashboard/create-customers.html>

Following is the folder structure - `/docs/components/`

Try to retrieve the hidden flag.

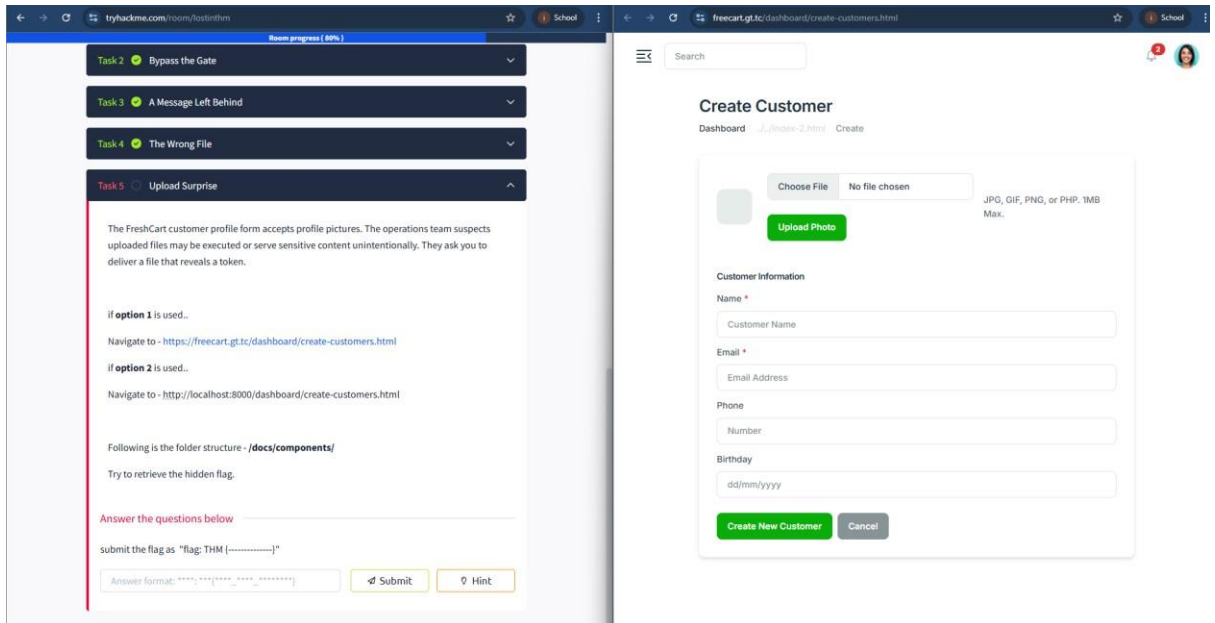
Answer the questions below

submit the flag as "flag: THM {-----}"

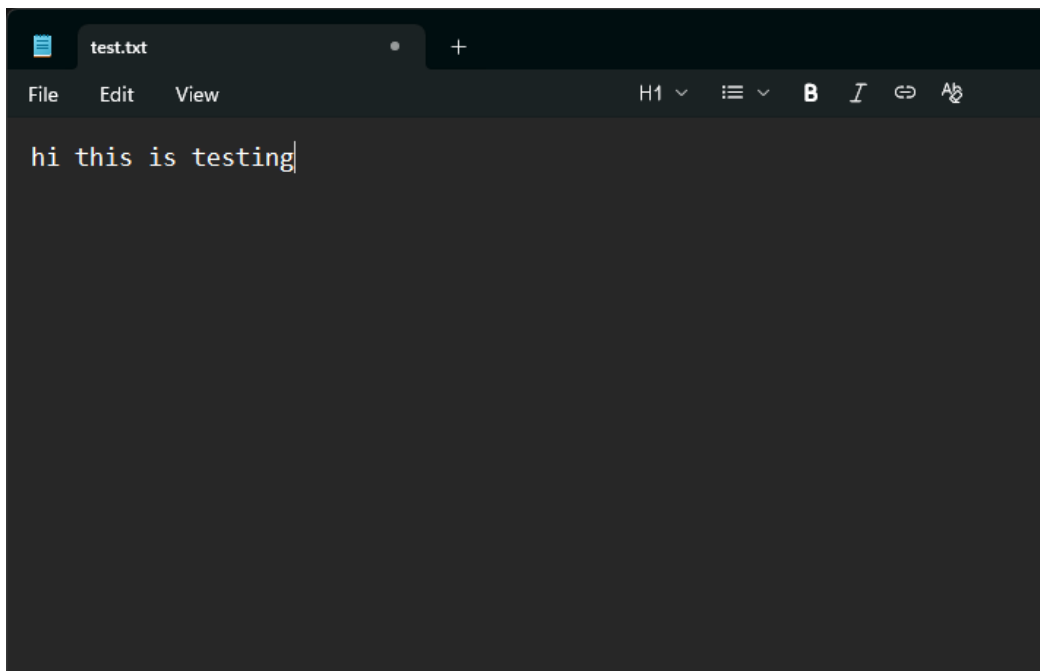
Answer format: ****; **{**** ** *****}

➤ Steps to Capture the Flag

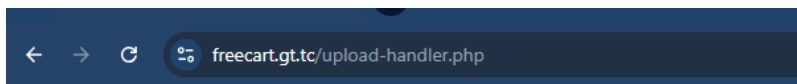
Step 1 – Navigate to the link provided.



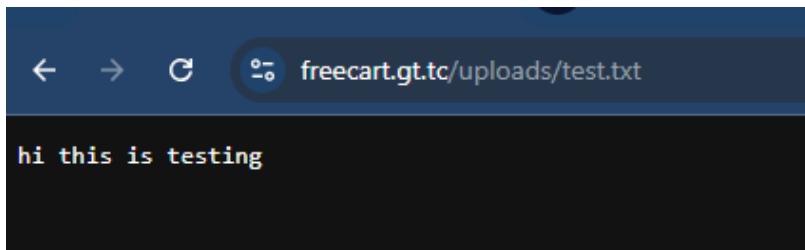
Step 2 – Try uploading a text file.



TryHackMe Walkthrough



File uploaded successfully! Access it at: uploads/test.txt

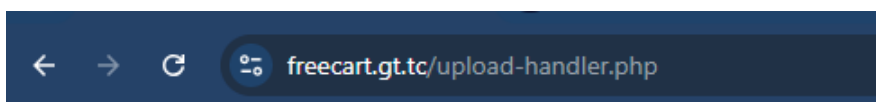


Here the text file has been uploaded successfully.

Step 3 – Try to access the file path given and display the content in the file.

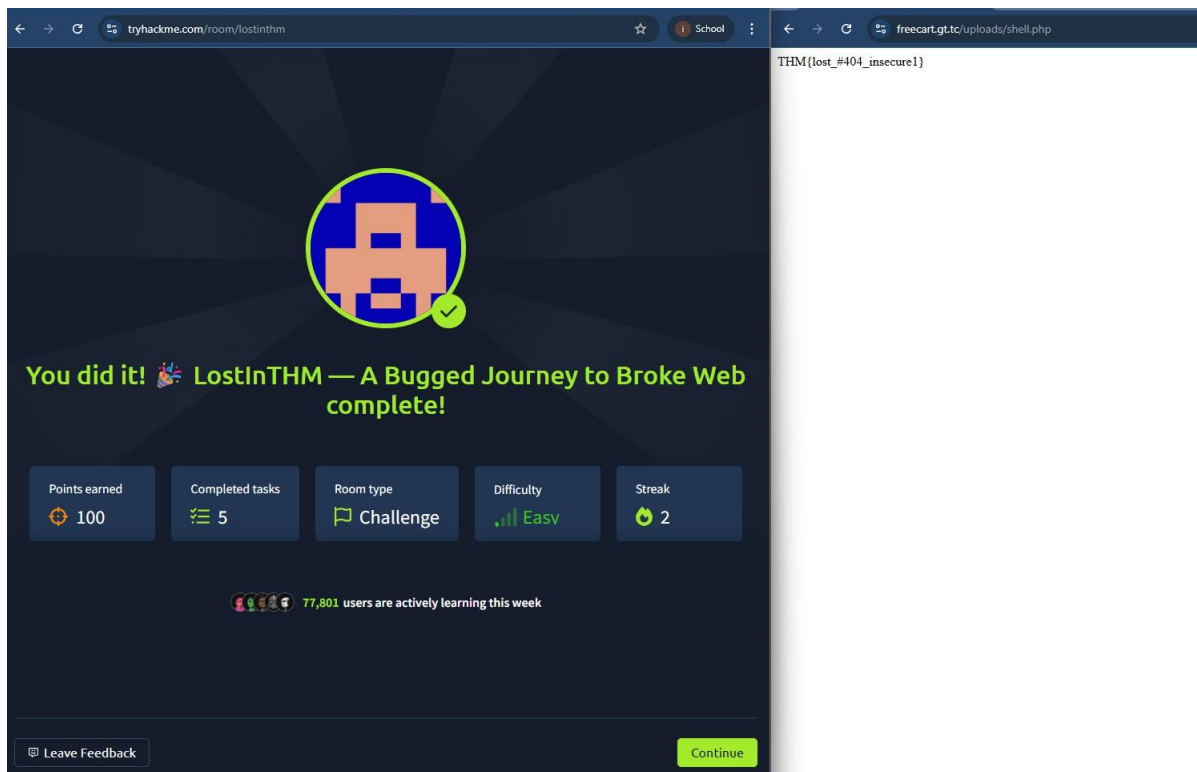
../docs/components

```
<?php echo file_get_contents('../docs/components/flag.txt'); ?>
```



File uploaded successfully! Access it at: uploads/shell.php

TryHackMe Walkthrough



➤ Mitigation techniques

- Validate the file type and its content before sending it to the server.
- Enforce an allowed file type list to avoid suspicious uploads.
- Store the uploaded files away from the root files.
- Remove execution permissions from upload directory.
- Implement a monitoring detection for suspicious uploads and executions.

4. Explanation of flags.

▪ Flag Format

flag: THM {_____}

Task - 1 flag (SQL injection)

flag: THM{LostInTHM_sqli}

Task - 2 flag (Cross-site scripting)

flag: THM{lost_xxs#14}

Task - 3 flag (Path Traversal)

flag: THM{PT_lostin\$13}

Task - 4 flag (Insecure Upload)

flag: THM{lost_#404_insecure}

▪ Learning outcome

- Get hands-on experience in identifying, understanding and exploiting a basic web vulnerability.
- Importance of proper mitigation and securing coding practices.
- Understand how vulnerabilities can be connected to escalate impact.
- To get a real-world scenario- based experience.