

Race Conditions 2

Please review the binder race conditions contents for instructional information, or Burp Suite to review more information regarding Burp. Please note this was done without the use of Wireshark as I am not experienced enough yet.

So the challenge for this particular aspect is we have 3 accounts, with 3 separate credentials.

Name: Rasser Cond

U: 4621

P: blueApple

Balance: \$100

Name: Zavodni Stav

U: 6282

P: whiteHorse

Balance: \$100

Name: Warunki Wycigu



U: 9317

greenOrange

Balance: \$100

The aim is to use these 3 accounts to amass more than \$1000 dollars. I will log into each account individually to identify how much money I have initially to work with. *I have updated them next to the names. So they all have \$100.

The IP: 10.10.155.137:5000

 Dashboard Rassar Cond 

Balance
\$100 USD

Favourite Transfers

Zavodni Stav

Warunki Wycigu

So from analysing the information at current we have 3 accounts all closely aligned as each other are all within the favourite transfers of each account. So the first thing I am going to do, is take a breathe and prepare Burp again with the same settings as outlined in Race Conditions 1, why? Because there is a small chance it is the same principle, plus we have acquired that knowledge in our brain and it's worth exploring what we already know!

So I am going to choose Rassar Cond to send Zavodni Stav \$10 and similarly to last time, acquire the same setup. This is by obtaining a POST request with the transfer and then sending it to the repeater. I will show how I achieved these steps.

Transfer Funds

You send

0.50 USD

Total fees

Zavodni Stav gets

Continue

Ooo Okay! We are working with Fees. I know it was too good to be true, the ones acquiring these fees are about to become rich on the account of vulnerabilities.

To make it even I will send \$5 and take it slow, this means what Zavodni Stav will receive will be a grand total of \$4.75. From this as well I can work out how much the tax rate actually is - %5 it seems.



Transaction Successful

\$4.75 was sent to Zavodni Stav. Let's go into the HTTP history and find that POST request.

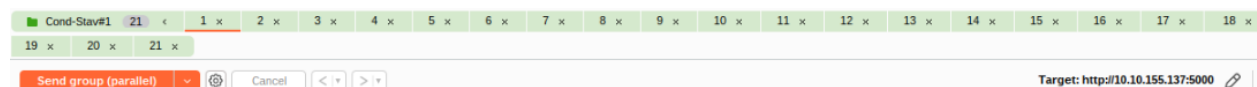
The screenshot shows a web browser's developer tools with the 'Request' tab selected. The request is a multipart/form-data submission to http://10.10.155.137:5000/transfer/6282. The request body contains three parts: 'fund_being_transferred' with value '5', 'calculatedfee' with value '0.25', and 'receiver_amount' with value '4.75'. The browser interface shows 'Cond-Stav#1' and a 'Send' button.

```
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://10.10.155.137:5000/transfer/6282
8 Content-Type: multipart/form-data;
  boundary=-----3748014363637402551897708481
9 Content-Length: 432
10 Origin: http://10.10.155.137:5000
11 Connection: keep-alive
12 Cookie: session=
  .eJwtjDsKgDAQBa8ir05hRCzSiTfw8quuKMRNyAcLyd0NaDcP5s0DYulgSed
  KiTeYFDIr5MjhlN3BPFjIkqwMo9tWQeiqiJliVZrJyQYFX9ftQrljsZlH7y3
  jq_x-P3QapbxbpaCUl.aH-ESQ.gPg7ZCCoXR9ep-wCQKNyLCKxmUw
13 Priority: u=0
14
15 -----3748014363637402551897708481
16 Content-Disposition: form-data; name="fund_being_transferred"
17
18 5
19 -----3748014363637402551897708481
20 Content-Disposition: form-data; name="calculatedfee"
21
22 0.25
23 -----3748014363637402551897708481
24 Content-Disposition: form-data; name="receiver_amount"
25
26 4.75
27 -----3748014363637402551897708481--
28
```

So here we have 3 parameters,

1. Fund being transferred
2. Calculatefee
3. Receiver_amount

Before I start playing with any of that, I am going to duplicate this request and see how much is sent when I parallel request this 20 times... For science.



Let's see what happens, I am more interested in seeing how the fees work, already I can see because there is more options than the last challenge, could introduce some interesting twists. But let's send it and see what the response is.

The attack box froze.. Again... Time to restart so I will essentially do the same thing again. This I can imagine happens often due to the volume of people active on it.

Right so it saved the amount I sent but because I refreshed the POST request isn't there.... Lost money to fees, I will send another \$5 and repeat the process again.

The screenshot displays a web browser window with a network request inspector open. The target URL is `http://10.10.155.137:5000`. The request is a POST method with a content type of `multipart/form-data`. The request body is a multipart form with the following parts:

- `Accept-Language: en-US,en;q=0.5`
- `Accept-Encoding: gzip, deflate, br`
- `Referer: http://10.10.155.137:5000/transfer/6282`
- `Content-Type: multipart/form-data; boundary=-----169826911327528867883243979834`
- `Content-Length: 440`
- `Origin: http://10.10.155.137:5000`
- `Cookie: session=.eJwTjME3gDAQBfUrfQdRLucH8x7s4N0ThXgJMcGH2LsB_c3C7MygGDwWsMBUeIEDPrJCPNvslroGxMZkpmhuyYvFTSOx8jptE42Wfmg4NK6rE9_TCZy75xhfJnfr9uqxP08iQULcA.aH-Jlw.h9HQ5E6CRzTaZnW5mfEqFa1sTlg`
- `Priority: u=0`
- `Content-Disposition: form-data; name="fund_being_transferred"`
- `5`
- `Content-Disposition: form-data; name="calculatedfee"`
- `0.25`
- `Content-Disposition: form-data; name="receiver_amount"`
- `4.75`

The response is empty. The web application interface shows a "Transfer Funds" form with the following fields:

- You send**:
- 0.25 USD**: Total fees
- Zavodni Stav gets**:
- Processing...**: A button with a loading spinner.

So I have the post request and the processing tab is still going! This might work to our advantage

```
1 HTTP/1.1 200 OK
2 Server: unicorn
3 Date: Tue, 22 Jul 2025 12:55:11 GMT
4 Connection: keep-alive
5 Content-Type: application/json
6 Content-Length: 16
7 Vary: Cookie
8 Set-Cookie: session=
  .eJwtjMEJgDAQBFuRfQdREch8xA7s4KInCvESYoIPsXcD-puF2b1BKW4scZ8
  p8gIdQ2KFdHLYZXQNwxZkpmh-6qsFISOzJjozE4x0lmg4P06XMh_GJt48N4
  yvszvt11T43leh7Ylaw.aH-KLw.UVfJubwB8Tel pNqpHy6EXdt0WU4;
  HttpOnly; Path=/
9
.0 {
  "result":true
}
.1
```

So it sent, time to log into Stavs account and see if it worked the way I hoped it would.

Balance

\$152.25 USD

From \$100 to \$152.25 I think it works the way we intended it to! We are learning and this is an exceptional insight. Perhaps the processing not being fully request was equally to blame? The way I interpret that is the server hadn't fully acknowledged this request only to receive another 20 requests simultaneously! Too much to deal with evidently. What I am going to do now is send another transfer from Stav but transfer it down to Wycigu.

Transfer Funds

You send

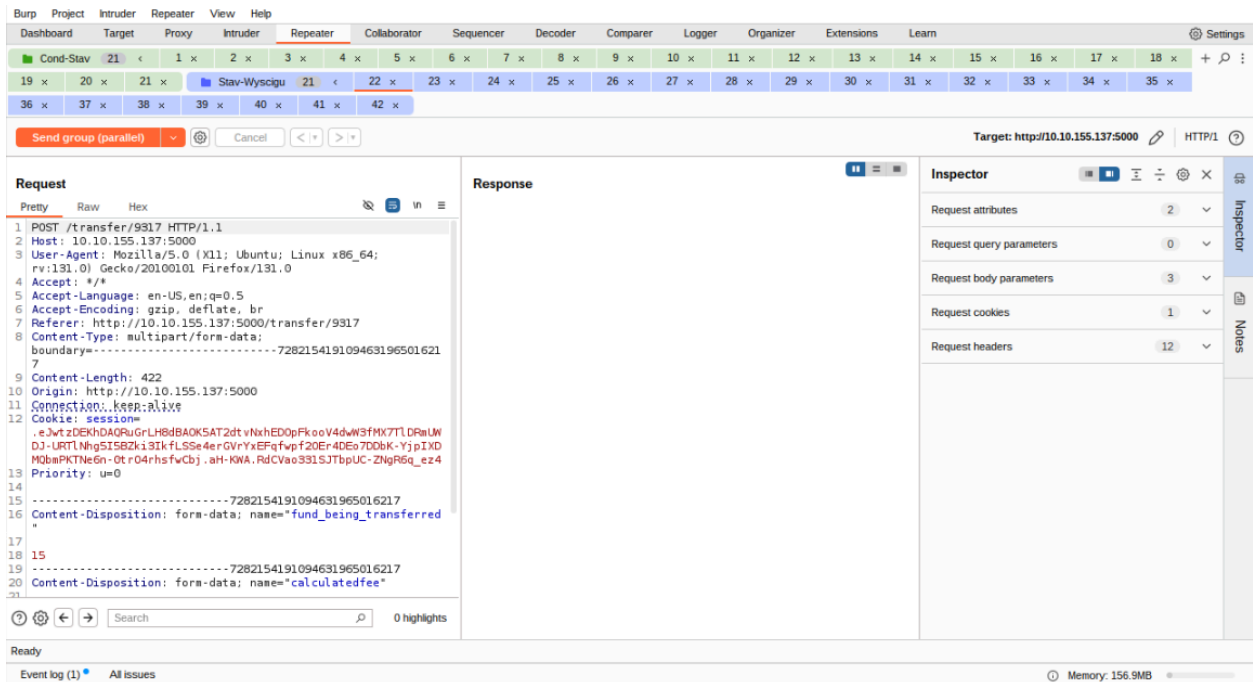
0.75 USD

Total fees

Warunki Wycigu gets

Continue

This time I am going to send \$15 paying the \$0.75 fee. This time the transaction successful message popped up, but I am still going to abuse the POST request. Let's find it in the HTTP history. Right click and send to repeater and repeat the same process again, adding another group of 20 tabs and naming it "Stav-Wycigu" so that way I can keep track of what processes have been done and what the post request is doing in simpler terms.




Here we go, the final stages before selecting “Send group (parallel)”. Let’s click it and see what happens.


```

1 HTTP/1.1 200 OK
2 Server: unicorn
3 Date: Tue, 22 Jul 2025 13:01:01 GMT
4 Connection: keep-alive
5 Content-Type: application/json
6 Content-Length: 16
7 Vary: Cookie
8 Set-Cookie: session=
  .eJwztZEEKgzAQRuGryL80hQ5YJCfovjt3Yxwx0E4kmehCvLtCXb7F-3ZwtVn
  UYmCTEd5yFYdaJEedEvyOgb-sQeCfRA9qHZR_V6HnNY0am4_xCoeFS9lSvgh
  sczR5plwEf-o-XtQRjuMEHusm4A.aH-LjQ.f3nl00nt7y-xRZRY9fahLhL4l
  GY; HttpOnly; Path=/
9
10 {
11     "result":true
  }

```

200 OK is a great indicator that it went through as well as seeing the result = true. So let’s now log into Wyscigu’s account and see what’s happened.

 Dashboard

Warunki Wycisgu 

Balance
\$214.0 USD

Favourite Transfers

Rasser Cond

Zavodni Stav

Fantastic, it appears that the processes are working exactly the way I was hoping. So now we are generating a health prize pool out of thin air and still paying the fees, which no doubt I could perhaps change, but to try and minimize our presence here, and tweaking too much I will achieve the \$1000 cash pool at this rate regardless.

So now, we will go full circle and Wycisgu will now send money to Conds account. This time I will send \$20, and see what happens when we duplicate the tabs this time.

Transfer Funds

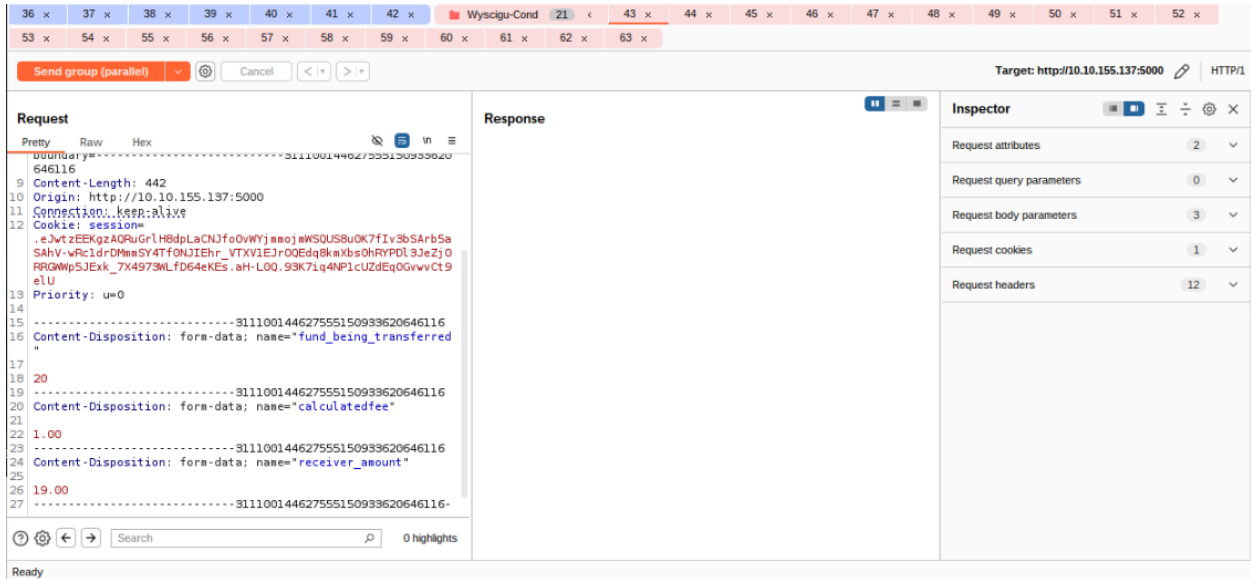
You send

1.00 USD

Total fees

Rasser Cond gets

Continue



I select send group again and to check that it went through correctly I will wait patiently for the response of the server.

```
1 HTTP/1.1 200 OK
2 Server: gunicorn
3 Date: Tue, 22 Jul 2025 13:05:48 GMT
4 Connection: keep-alive
5 Content-Type: application/json
6 Content-Length: 16
7 Vary: Cookie
8 Set-Cookie: session=
  .eJwtzEEKgzAQRuGryL80UrEgZW6dD1NxxjajmWSoUjw7hVO-RbvqyArM0t
  JgQo_4Ysa0lhmTTIt8BUPEpMEhu-Ga3txEPsgZHU5JWacc0hRYPDL3L-Lbo
  jiMosdyWJjEM7r1vfDdi2P6-PKFA.aH-MrA.dl4speHdQBZtTyKaew7lNPfq
  _30; HttpOnly; Path=/
9
10 {
11   "result":true
12 }
```

Fantastic sign, I will not log into Conds account again to see the result of this request overload.

Balance
\$294.0 USD

Favourite Transfers

Zavodni Stav	Warunki Wycigu
--------------	----------------

Fantastic, so we are generating more money. What I will now do, is try manipulating the requests again similarly to how I did it in Race Conditions 1, even though we sent \$1.50 in some requests I would manually change the POST request value and it would still generate incredible results, let's play around with not only the value but the fees too, maybe we can generate an error message? Anything at this point is extra, because we have a system that works and trying new things outside of the norm generates little risk, so let's play around.

I will send money to Stav again, let's make it \$5 on paper, the transfer

Transfer Funds

You send

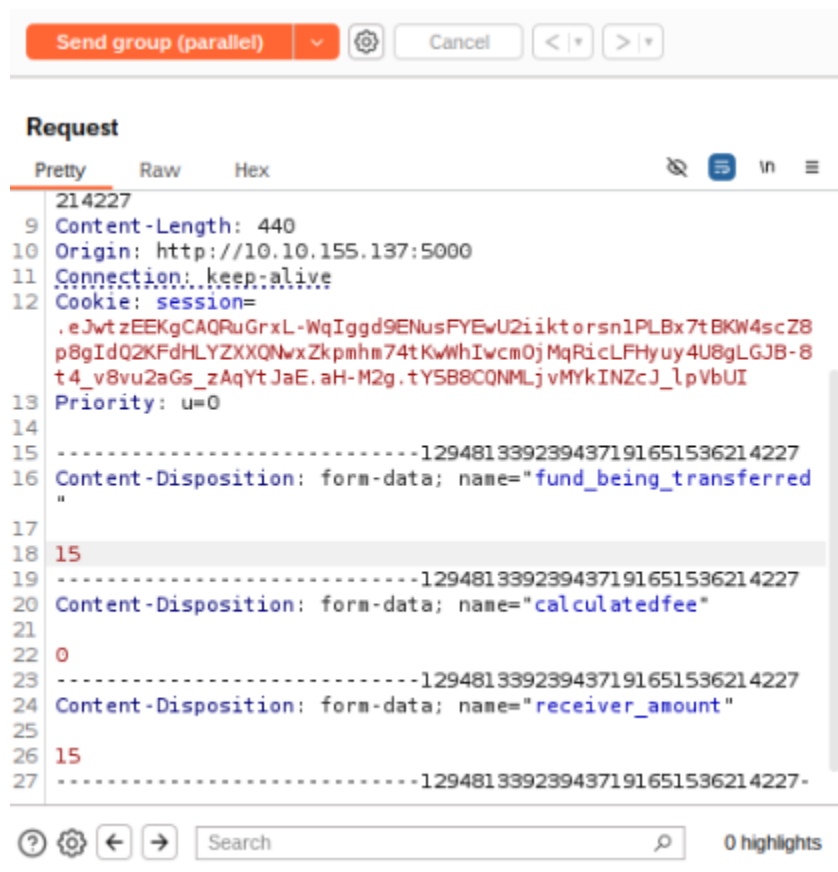
0.25 USD

Total fees

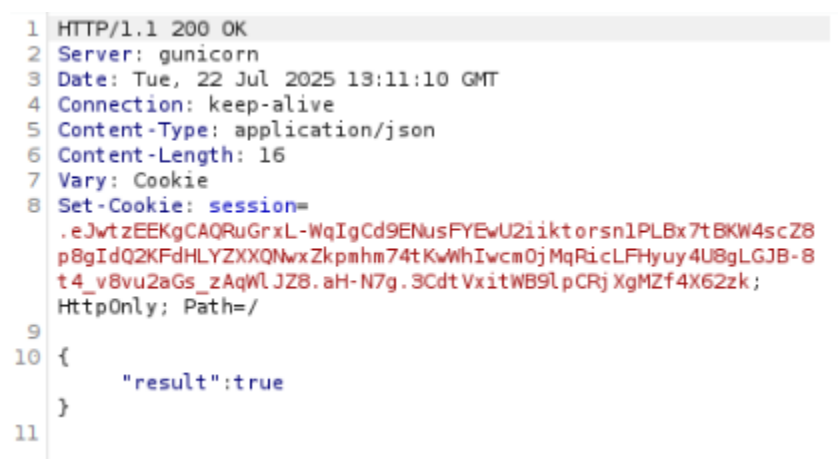
Zavodni Stav gets

Continue

But this time I am going to manipulate the POST request more, as well as duplicating the request, think of it like a double - joint attack.



So this time I have set up my tabs as per usual, but I have set the parameters manually, such as the funds being transferred to \$15, the fee is \$0 and the receiver amount is \$15... Now I could experiment and set this to a billion, but to try keep it realistic and avoid creating errors, let's see if this does work.



Right so everything looks okay from this perspective but the real results are shown from the account balance so let's log into Stavs account and see.

Balance
\$232.0 USD

Favourite Transfers

Rasser Cond

Warunki Wycigu

Interesting, now it is possible I got confused along the way, taking control of 3 accounts can become confusing, this time I will send funds from Stavs account to Wycigu again and see what mischief I can perform.

Transfer Funds

You send

15

0.75 USD

Total fees

Warunki Wycigu gets

14.25

Continue

Target: http://10.10.155.137:5000

Request

```

9 Content-Length: 438
10 Origin: http://10.10.155.137:5000
11 Connection: keep-alive
12 Cookie: session=.eJwztDE0gzAQbdGroF9bEVCE48PQp6NbYCNbgnVkr6FAvnuQSDnFvBOU1bGonOl5gdWY2SAnjl4-AfbERCvJzLBN3z1qA6HtCoy0hOV89VbaYfCl1I4QLwGH88pDiIlxs_j2b5a1PIDAuMmtQ.aH-PVA.SA8x5cmgZwa9xjaIxUj inj xK9Os; HttpOnly; Path=/
13 Priority: u=0
14
15 -----12706752951911440685226090108
16 Content-Disposition: form-data; name="fund_being_transferred"
17
18 20
19 -----12706752951911440685226090108
20 Content-Disposition: form-data; name="calculatedfee"
21
22 1
23 -----12706752951911440685226090108
24 Content-Disposition: form-data; name="receiver_amount"
25

```

Response

```

1 HTTP/1.1 200 OK
2 Server: gunicorn
3 Date: Tue, 22 Jul 2025 13:17:08 GMT
4 Connection: keep-alive
5 Content-Type: application/json
6 Content-Length: 16
7 Vary: Cookie
8 Set-Cookie: session=.eJwztDE0gzAQbdGroF9bEVCE48PQp6NbYCNbgnVkr6FAvnuQSDnFvBOU1bGonOl5gdWY2SAnjl4-AfbERCvJzLBN3z1qA6HtCoy0hOV89VbaYfCl1I4QLwGH88pDiIlxs_j2b5a1PIDAuMmtQ.aH-PVA.SA8x5cmgZwa9xjaIxUj inj xK9Os; HttpOnly; Path=/
9
10 {
11     "result": true
12 }

```

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 3
- Request cookies: 1
- Request headers: 12

This time I manually changed the values and in accordance I also set the fee limit just in case, as well as the received amount to the appropriate form as with the deduction of the fee, let's see what happens. I have also increased the amount of requests to 30 instead. Now I am hoping to see that the value goes beyond \$300 which will prove that it is in fact working as it goes beyond what all the accounts had combined at the beginning. I have no doubt that this isn't working but it's just a polite reminder that this is in fact certainly working.

```

1 HTTP/1.1 200 OK
2 Server: gunicorn
3 Date: Tue, 22 Jul 2025 13:17:08 GMT
4 Connection: keep-alive
5 Content-Type: application/json
6 Content-Length: 16
7 Vary: Cookie
8 Set-Cookie: session=.eJwztDE0gzAQbdGroF9bEVCE48PQp6NbYCNbgnVkr6FAvnuQSDnFvBOU1bGonOl5gdWY2SAnjl4-AfbERCvJzLBN3z1qA6HtCoy0hOV89VbaYfCl1I4QLwGH88pDiIlxs_j2b5a1PIDAuMmtQ.aH-PVA.SA8x5cmgZwa9xjaIxUj inj xK9Os; HttpOnly; Path=/
9
10 {
11     "result": true
12 }

```

Fantastic, let's log into Wyscigu's account and see what the total amount is now.

Balance
\$510.25 USD

Favourite Transfers

Rasser Cond	Zavodni Stav
-------------	--------------

Fantastic, so this is evident beyond comprehension that this is working very very well. I am actually smiling because this it shows that the knowledge accumulated from previous challenges are really shining here, with no need to look back or review notes, it's just trial and error. So let's continue the cycle and send a transfer this time of \$100 to Conds account following the same steps but this time I won't manipulate the actual contents, just do it as intended.

Transfer Funds

You send

5.00 USD

Total fees

Rasser Cond gets

Continue

The fees are starting to add up! \$5... If only we didn't have infinite money this would be a problem. So let's see what we can do here now.

Target: http://10.10.155.137:5000 | HTTP/1

Request

Raw

```

SANWe4Ys8U1HwI XJT4DdM9CLJDD_0x-400A19WZEkXmWw_JNIUMDw4dyXhd
tCqIyy01JiuPH_a_rub-g1h30Ay1B.aH-Pfg.WohA4sXUFSLOA3dhEsQULKQ
nuco
Priority: u=0
-----176879733117291031732735350671
Content-Disposition: form-data; name="fund_being_transferred"
17
18 1.00
-----176879733117291031732735350671
Content-Disposition: form-data; name="calculatedfee"
21
22 5.00
-----176879733117291031732735350671
Content-Disposition: form-data; name="receiver_amount"
24
25 95.00
26

```

Response

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 3
- Request cookies: 1
- Request headers: 12

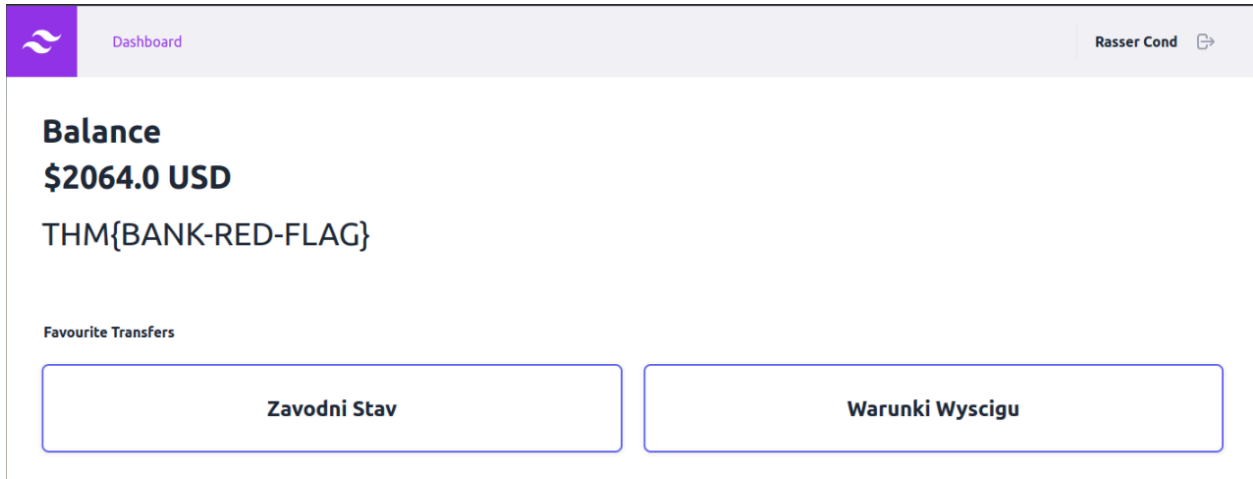
A small part of me feels like an error is incoming, but I want to see what happens here, this is a very large value, requesting 30 times, it might be that the money in our account cannot sustain this sort of request but let's see, there is only 1 way of finding out.

```

1 HTTP/1.1 200 OK
2 Server: gunicorn
3 Date: Tue, 22 Jul 2025 13:21:33 GMT
4 Connection: keep-alive
5 Content-Type: application/json
6 Content-Length: 16
7 Vary: Cookie
8 Set-Cookie: session=
  .eJwtzEEKwjAQRuGryL80xVqKmEu47HqajjGoU5lkECmSew06fIv3bSArd5a
  SAhVe4Isa01hmTXJb4TfM9CQJDD_0x-400gi9WmEiNXmkw_TNIUMDw5ty_qz
  aFERllquSRMaP-1-XoT-j1h3Nayh_.aH-QXQ.cyieu7Nn25xxMWqZ37S0Zc7
  _7K4; HttpOnly; Path=/
9
10 {
11   "result":true

```

😬 This means that it did work, with no error generated, now we log into Conds account and see if we achieved the \$1000 required to pass the challenge.



We have successfully done it and got my sweet, sweet flag. THM{BANK-RED-FLAG}. So let's summarize the end values.

Name: Rasser Cond

U: 4621

P: blueApple

Balance: \$2064

Name: Zavodni Stav

U: 6282

P: whiteHorse

Balance: \$157

Name: Warunki Wycigu

U: 9317

greenOrange

Balance: \$110.25

So in the end everyone profited! But Cond won the lottery! I am sure he will split the winnings with his pals.

Conclusion:

So I have learnt that this knowledge can really stick once you repeat the processes over and over again, but also understand why it is occurring, not just doing it blindly. Why we place the POST request in the repeater to use Burp Suite to send the request quickly, multiple times to over burden the web servers ability to handle the requests, causing it to process the requests multiple times without being able to update it quick enough, making us create higher values that what we had initially. This vulnerability shows how tools can really amplify attacks.

