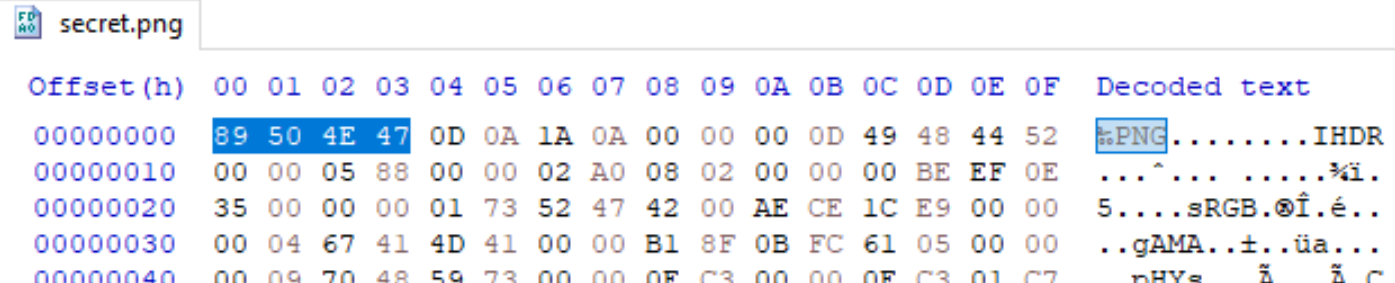
**BlackBox 2**

1. First, when I opened the folder I realized that there are 2 files which should be extracted but I don’t have their password so may I find them in the next stages.
2. Second, I tried to open the image file **secret.png** using a basic images program(like **Photos** on windows platform) but the file didn’t open.
3. Then I opened the image using hex editor (I used **HxD**) and found out that the **magic number** of the **png** format file was wrong, so I changed it to the right one and it worked! You can see that in the following image:



The image is:



1. In the image I saw a combination of characters **ODQxMjM0==** which looked like an encoded text in **UTF-8** encryption. Therefore, I wrote this script in python using **base64** library to decrypt it:

import base64

def decrypt(s:str):

    # The function gets a string.

    # The function returns the string after decrypting it.

    return base64.b64decode(s.encode()).decode()

print(decrypt("ODQxMjM0=="))

The output: **841234**

1. Then, I thought about what to do with this number. I tried using it as the password to check if it’s possible to extract the **secret.rar** folder and it worked!
2. Afterwards, I ran the **secret.exe** file using cmd which printed this text:

**>secret.exe**

**This software was valid up until 2011-11-11 11:11:11**

This output led me to change my machine’s time to a date before that: **2011-11-11 11:11:11**.

1. I run **secret.exe** again and it printed this:

**>secret.exe**

**Date Validation was previously done. Skipping condition check.**

**Current license value: a43fa**

**The current license registry value is invalid.**

**Expected that the sum of the hex values should be above 0x50 and also can also can be divived by 3 without a reminder.**

**Traceback (most recent call last):**

**File "secret.py", line 57, in <module>**

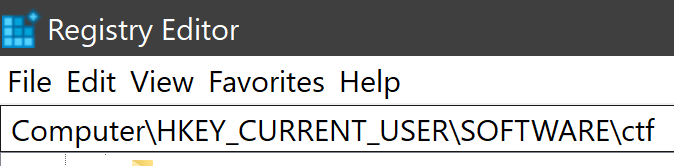
**File "secret.py", line 42, in check\_licence\_registry**

**NameError: name 'exit' is not defined**

**[33332] Failed to execute script 'secret' due to unhandled exception!**

Now I figured that it worked because of the message: “**Date Validation was previously done**”. In addition, I found that the **current license value** that showed, “**a43fa**”, is invalid because it’s hex value should be **above 0x50 and dividable by 3 without a reminder.**

1. Therefore, I tried to change this **current license value** to another value which follows the requirements (like: “**fffffffffffff**”) using the registry in the following path(I found it by using the **find** function with the “**a43fa”** value):



1. After I ran **secret.exe** again surprisingly the program started running a server by this output:

**>secret.exe**

**Date Validation was previously done. Skipping condition check.**

**Current license value: fffffffffffff**

**The current license is valid.**

**\* Serving Flask app 'secret'**

**\* Debug mode: on**

**Starting login server on port 8080...**

**WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.**

**\* Running on http://127.0.0.1:5000**

**Press CTRL+C to quit**

**\* Restarting with stat**

**Date Validation was previously done. Skipping condition check.**

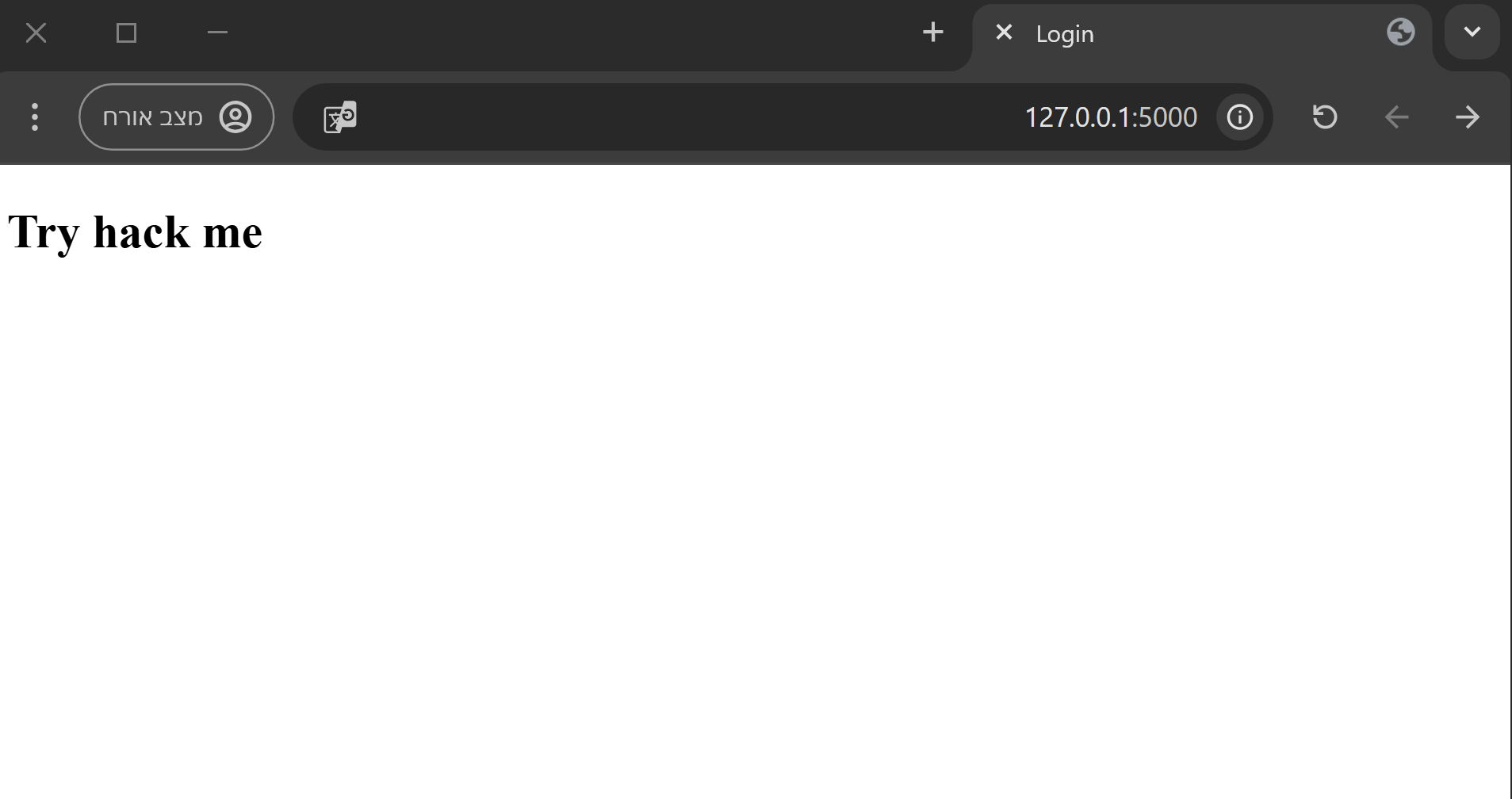
**Current license value: fffffffffffff**

**The current license is valid.**

**\* Debugger is active!**

**Starting login server on port 8080...**

**\* Debugger PIN: 130-702-582**

1. From this message I found some things: the **license** value is valid and the program running an http server which run on the **localhost** with the port **5000**. Therefore, I open a browser and entered the url of the server which is: “**http://127.0.0.1:5000**”. An image from the website: 
2. In the inspector I found this **JavaScript** script:

<script>

const x = window.location.href.split("\x69\x68\x61\x63\x6b\x79\x6f\x75");

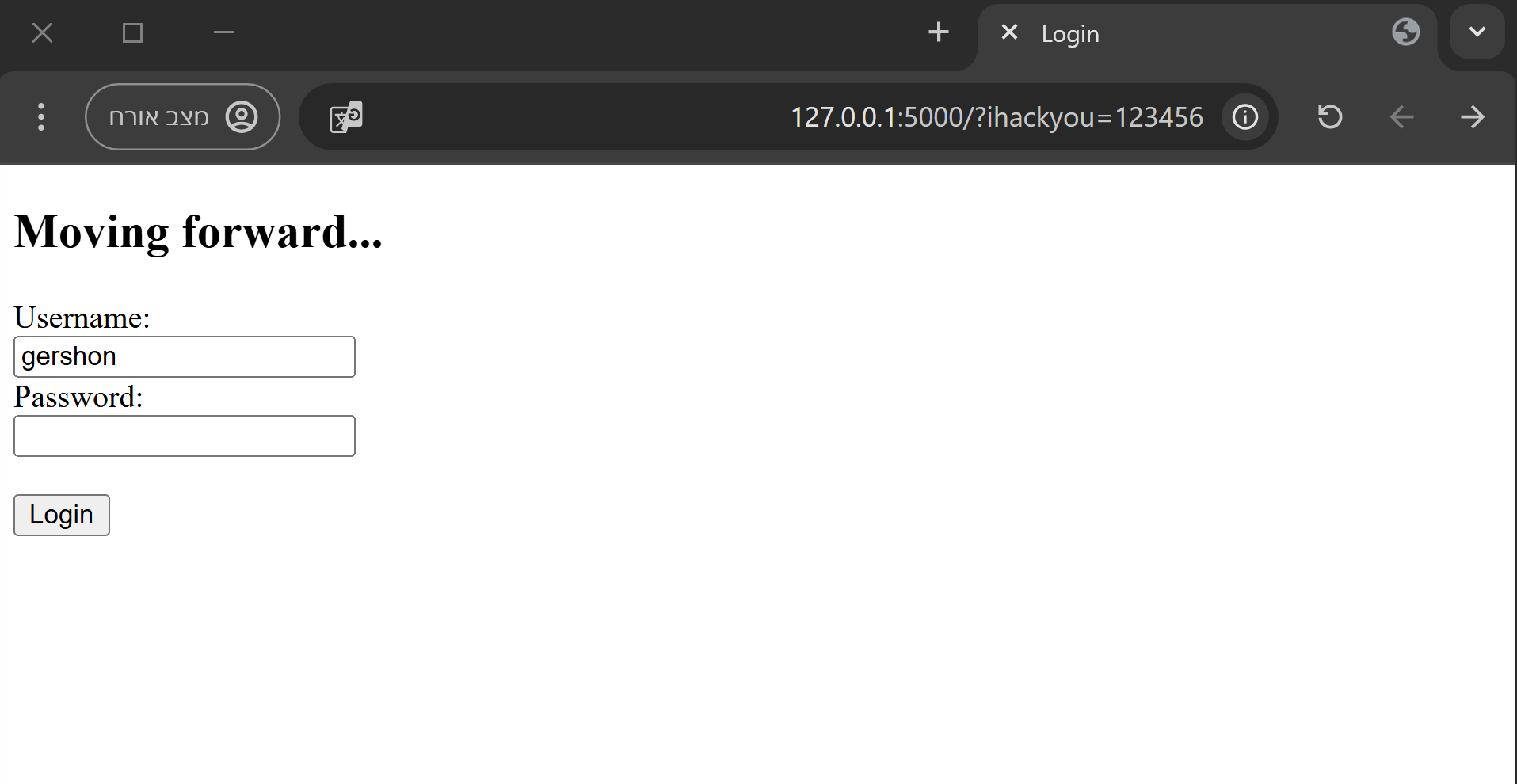
if (x.length === 1) {

console.log("You are missing something");

}

</script>

Then I went to the **Console** of the inspector and entered this **JavaScript** command to understand the string value : **> console.log("\x69\x68\x61\x63\x6b\x79\x6f\x75");** The output of this command was: “**ihackyou**”. Then I understood from the **JavaScript** script that probably something different will happen if the url contain the substring “**ihackyou**”.

1. Now, I add a simple unimportant parameter is the url with the name “**ihackyou**” and add it a random value and the website assigned a new page which seems like a login page: 
2. I have noticed that the username field already contains a defualt value which is: “**gershon**”. Then I tried to implement sql injection(like:”**1' OR 1=1**”) on the password field and it worked!
3. Now a new page was loaded on the website:

תמונה שמכילה טקסט, צילום מסך, חשמל, מולטימדיה

התיאור נוצר באופן אוטומטי

1. On this page, in the inspector, I found this JS script:

const urlParams = new URLSearchParams(window.location.search);

const a = urlParams.get("\x69\x68\x61\x63\x6b\x79\x6f\x75");

function \_\_das213csdas43(\_\_321dsa) {

    return 3 + \_\_321dsa -  parseInt("\x01".charCodeAt(0), 16)

}

const y = \_\_das213csdas43(3)

if (a && a  == "\x63\x79\x62\x72\x65\x72".split("\x72")[(Math.sin(Math.PI / 2) \*\* 2) + (Math.cos(0) \*\* 2) - 1 + y - 5]) {

    document.write('good');

    console.log("good job")

    window.location.href = "\x79\x6f\x75\x61\x72\x65\x63\x6c\x6f\x73\x65" + a

} else {

    document.write('Think harder now');

}

After analyzing it, I understood that “**x63\x79\x62\x72\x65\x72**” means “**x63ybrer**” and “**\x79\x6f\x75\x61\x72\x65\x63\x6c\x6f\x73\x65**” means “**youareclose**”.

In addition, I found that “**Math.cos(0) \*\* 2) - 1 + y - 5]**” **= 0**

“**Math.sin(Math.PI / 2) \*\* 2**” = **1**.

|||

\_|||\_

\|||/

\/

"**\x63\x79\x62\x72\x65\x72".split("\x72")[(Math.sin(Math.PI / 2) \*\* 2) + (Math.cos(0) \*\* 2) - 1 + y - 5])**” = “**e**”.