Challenge: Emdee five for life

Challenge Description:

Can you encrypt fast enough?

Context:

- Sending encrypted Data via Post Request
    - Getting the needed encryption type
    - How to send data
    - Is it fast enough

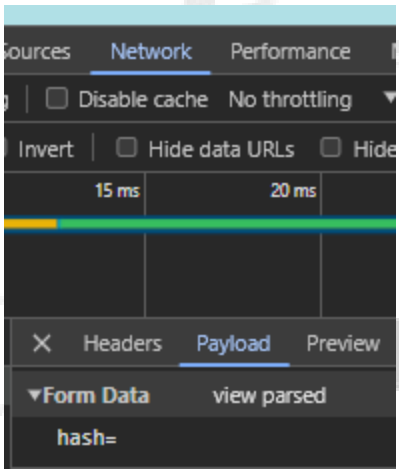
Notes:

- The encryption type is MD5 and the response to send is basically instantaneous

Challenge:

- First when when we launch the Instance for the web-page we see,

- After looking at this we can determine that we need to encrypt the random generated string into md5



- We will then need send it back via a POST request inspecting the web-page the POST request is "hash="

Continuation:

- To-do this I used a python script as it's one of the most simple and easiest to learn languages, so for this I needed to grab the string to encrypt to then send back to it via POST request to hopefully retrieve a Flag.
    - I began by setting up a Python session to initiate a GET request to fetch the initial HTML content from the webpage. The specific string of interest was located within the <h3> tag, which I then filtered and extracted from the server's response. After extracting the string, I applied the MD5 algorithm to hash it. Subsequently, I sent this hashed value back to the server via a POST request. The server's response, which should include the flag, would then be received and displayed.

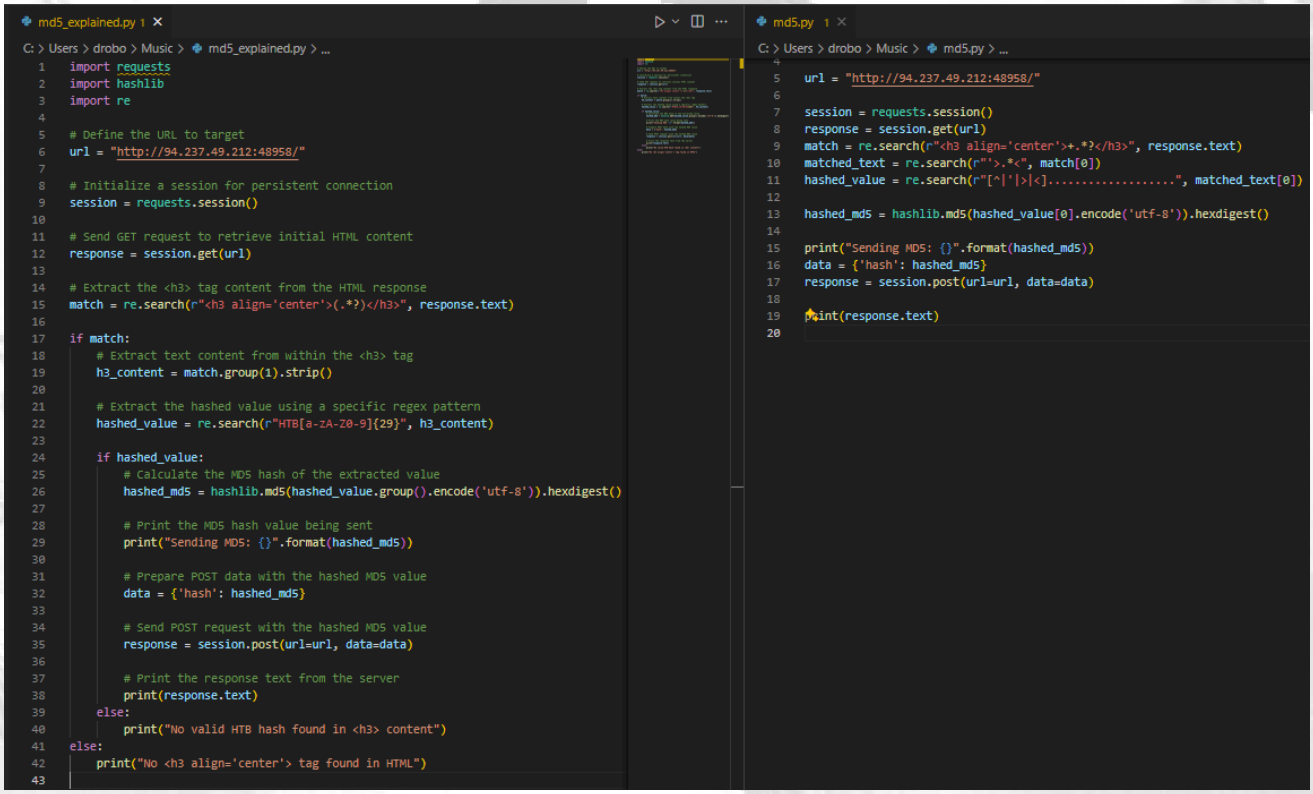
Flag:

HTB{N1c3_ScrIpt1nG_B0i!}

```
SKTOP-0ENQDDA:/mnt/c/Users/drobo/Music$ python3 md5.py
MD5: 92daa5e64364d10406b01fc320edd043


mdee five for life</title>

yle="background-color:powderblue;">
n='center'>MD5 encrypt this string</h1><h3 align='center'>uQe5hbMB7EH6x1TUuSxM</h3><p align='center'>HTB{N1c3_ScrIpt1nG_B0i!}<
er><form action="" method="post">
ype="text" name="hash" placeholder="MD5" align='center'></input>

ype="submit" value="Submit"></input>
/center>
```

The Script:

- The python script to complete this challenge will be on my github, as well as linked here, i will have a version that just contains the src code, i will also have a version that explains what each line does.
  - Normal
  - Explained



```python
import requests
import hashlib
import re

# Define the URL to target
url = "http://94.237.49.212:48958/"

# Initialize a session for persistent connection
session = requests.session()

# Send GET request to retrieve initial HTML content
response = session.get(url)

# Extract the <h3> tag content from the HTML response
match = re.search(r"<h3 align='center'>(.*?)</h3>", response.text)

if match:
    # Extract text content from within the <h3> tag
    h3_content = match.group(1).strip()

    # Extract the hashed value using a specific regex pattern
    hashed_value = re.search(r"HTB[a-zA-Z0-9]{29}", h3_content)

    if hashed_value:
        # Calculate the MD5 hash of the extracted value
        hashed_md5 = hashlib.md5(hashed_value.group().encode('utf-8')).hexdigest()

        # Print the MD5 hash value being sent
        print("Sending MD5: {}".format(hashed_md5))

        # Prepare POST data with the hashed MD5 value
        data = {'hash': hashed_md5}

        # Send POST request with the hashed MD5 value
        response = session.post(url=url, data=data)

        # Print the response text from the server
        print(response.text)
    else:
        print("No valid HTB hash found in <h3> content")
else:
    print("No <h3 align='center'> tag found in HTML")
```

```python
url = "http://94.237.49.212:48958/"

session = requests.session()
response = session.get(url)
match = re.search(r"<h3 align='center'>+.*?</h3>", response.text)
matched_text = re.search(r"'>.*<", match[0])
hashed_value = re.search(r"[^|'|>|<]...................", matched_text[0])

hashed_md5 = hashlib.md5(hashed_value[0].encode('utf-8')).hexdigest()

print("Sending MD5: {}".format(hashed_md5))
data = {'hash': hashed_md5}
response = session.post(url=url, data=data)

print(response.text)
```