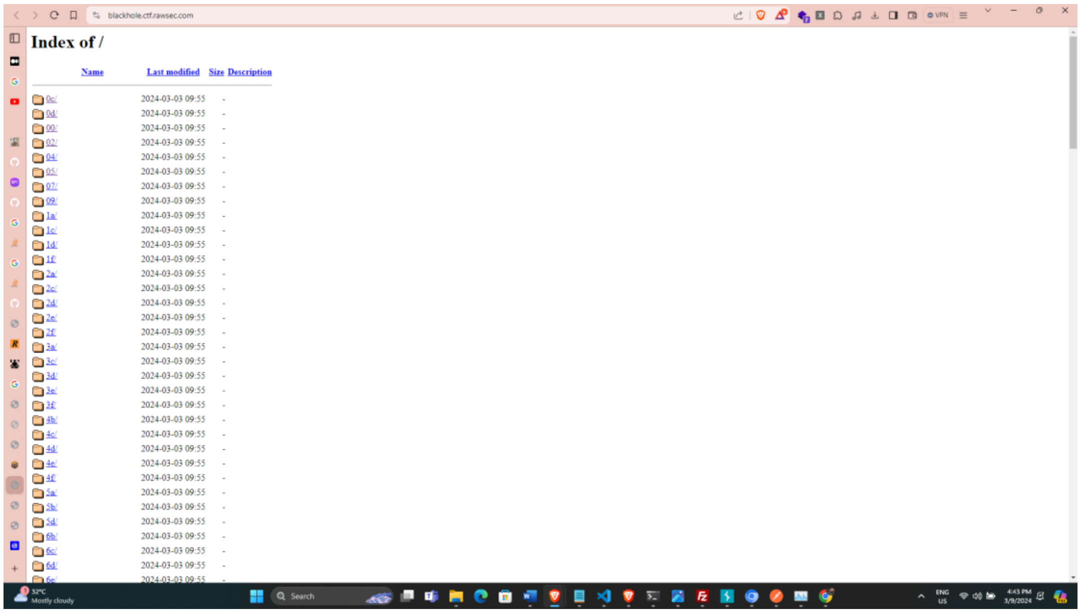
BlackHole (Web Challenge)

Team: JerungBiru

1. Firstly, we noticed that this is most likely same as the previous preliminary round challenge. The difference is now it does not use any emoji, instead it uses some kind of hex.



blackhole.ctf.rawsec.com

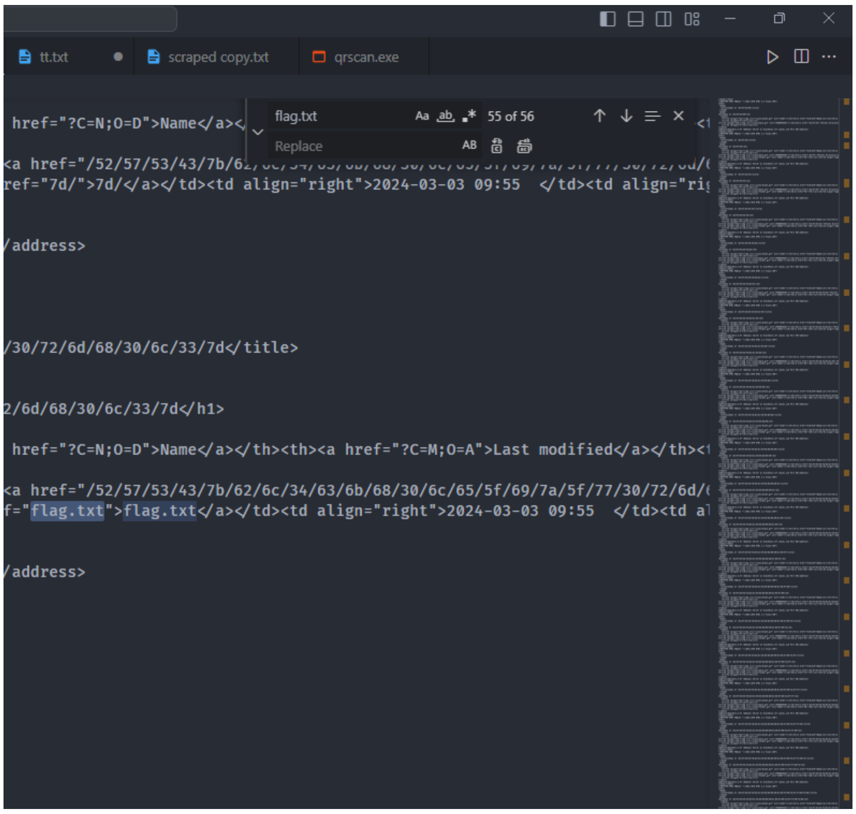
Index of /

```
Name   Last modified      Size Description
0c/    2024-03-03 09:55   -
0d/    2024-03-03 09:55   -
00/    2024-03-03 09:55   -
02/    2024-03-03 09:55   -
04/    2024-03-03 09:55   -
05/    2024-03-03 09:55   -
07/    2024-03-03 09:55   -
09/    2024-03-03 09:55   -
1a/    2024-03-03 09:55   -
1c/    2024-03-03 09:55   -
1d/    2024-03-03 09:55   -
1f/    2024-03-03 09:55   -
2a/    2024-03-03 09:55   -
2c/    2024-03-03 09:55   -
2d/    2024-03-03 09:55   -
2e/    2024-03-03 09:55   -
2f/    2024-03-03 09:55   -
3a/    2024-03-03 09:55   -
3c/    2024-03-03 09:55   -
3d/    2024-03-03 09:55   -
3e/    2024-03-03 09:55   -
3f/    2024-03-03 09:55   -
4b/    2024-03-03 09:55   -
4c/    2024-03-03 09:55   -
4d/    2024-03-03 09:55   -
4e/    2024-03-03 09:55   -
4f/    2024-03-03 09:55   -
5a/    2024-03-03 09:55   -
5b/    2024-03-03 09:55   -
5d/    2024-03-03 09:55   -
6b/    2024-03-03 09:55   -
6c/    2024-03-03 09:55   -
6d/    2024-03-03 09:55   -
6e/    2024-03-03 09:55   -
```

2. We used the same script that we have used in the preliminary round.

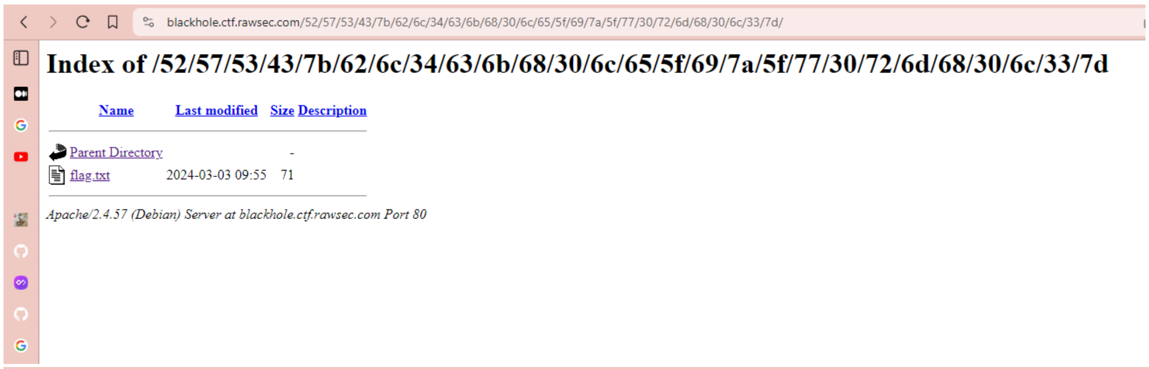
```python
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin

visited = set()

def get_directory_content(url, output_file):
    try:
        # Send a GET request to the URL
        response = requests.get(url)
        response.raise_for_status()  # Raise an error for bad responses

        # Parse the HTML content using BeautifulSoup
        soup = BeautifulSoup(response.text, 'html.parser')

        # Print the content of the current page to the terminal
        print(response.text)

        # Save the content to the output file
        with open(output_file, 'a', encoding='utf-8') as file:
            file.write(response.text)

        # Check if "RWSC" is present in the content
        if "RWSC{" in response.text:
            print("Found 'RWSC' in:", url)
            return

        # Find all directory links under the "Directories" section
        directory_links = soup.select('tr > td > a')

        # Follow each directory link recursively
        for directory_link in directory_links:
            absolute_url = urljoin(url, directory_link['href'])
            if absolute_url not in visited:
                visited.add(absolute_url)
                get_directory_content(absolute_url, output_file)

    except requests.exceptions.RequestException as e:
        print(f"Error accessing the website: {e}")

# Starting URL
start_url = "https://blackhole.ctf.rawsec.com/"

# Output file
output_file = "scraped.txt"

# Call the function with the starting URL and output file
get_directory_content(start_url, output_file)

```

3. Unfortunately, we did not find any text that starts with RWSC. However, we did log all the outputs into a *scraped.txt* file. Let's check the file.
4. From the *scraped.txt* file, we have found that there are many distractions by the HTML tags of "IAM flag.txt". We get rid of all those HTML tags and now we only left with 56 occurrences of flag.txt.



```
 href="?C=N;O=D">Name</a><
<a href="/52/57/53/43/7b/62
ref="7d/">7d/</a></td><td align="right">2024-03-03 09:55  </td><td align="ri

/address>



/30/72/6d/68/30/6c/33/7d</title>


2/6d/68/30/6c/33/7d</h1>

 href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><

<a href="/52/57/53/43/7b/62/6c/34/63/6b/68/30/6c/65/5f/69/7a/5f/77/30/72/6d/
f="flag.txt">flag.txt</a></td><td align="right">2024-03-03 09:55  </td><td al


/address>
```

5. We have found this particular HTML tag to be super interesting. We navigated to this URL and unfortunately not the flag that we are searching for.

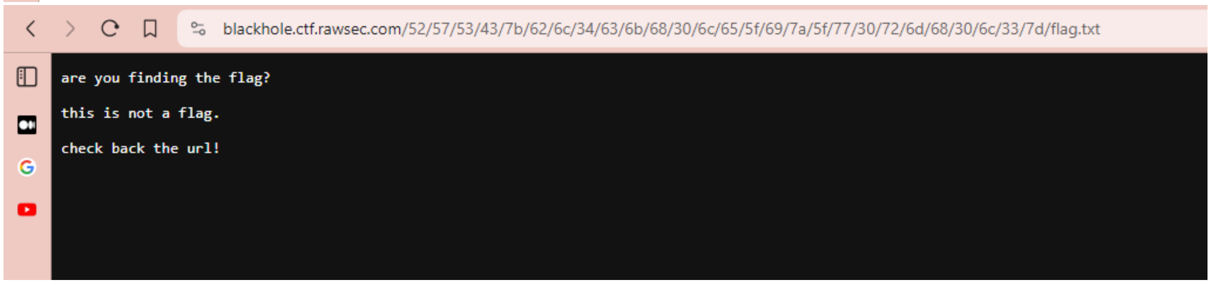


blackhole.ctf.rawsec.com/52/57/53/43/7b/62/6c/34/63/6b/68/30/6c/65/5f/69/7a/5f/77/30/72/6d/68/30/6c/33/7d/

Index of /52/57/53/43/7b/62/6c/34/63/6b/68/30/6c/65/5f/69/7a/5f/77/30/72/6d/68/30/6c/33/7d

Name	Last modified	Size	Description
Parent Directory		-	
flag.txt	2024-03-03 09:55	71	

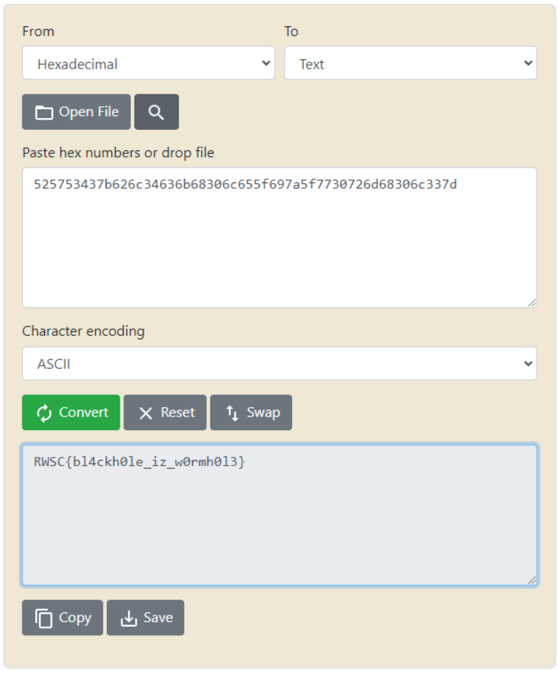
Apache/2.4.57 (Debian) Server at blackhole.ctf.rawsec.com Port 80



blackhole.ctf.rawsec.com/52/57/53/43/7b/62/6c/34/63/6b/68/30/6c/65/5f/69/7a/5f/77/30/72/6d/68/30/6c/33/7d/flag.txt

```
are you finding the flag?

this is not a flag.

check back the url!
```

6. However, it tells us that we must check the URL. Through some inspections, we found that the URL resembles some kind of ASCII representation of characters in hexadecimal.
7. We navigated to https://www.rapidtables.com/convert/number/hex-to-ascii.html to translate the hexadecimals into a readable string and woilah!



From: Hexadecimal
To: Text

Paste hex numbers or drop file

```
525753437b626c34636b68306c655f697a5f7730726d68306c337d
```

Character encoding: ASCII

```
RWSC{bl4ckh0le_iz_w0rmh0l3}
```

Flag: **RWSC{bl4ckh0le_iz_w0rmh0l3}**