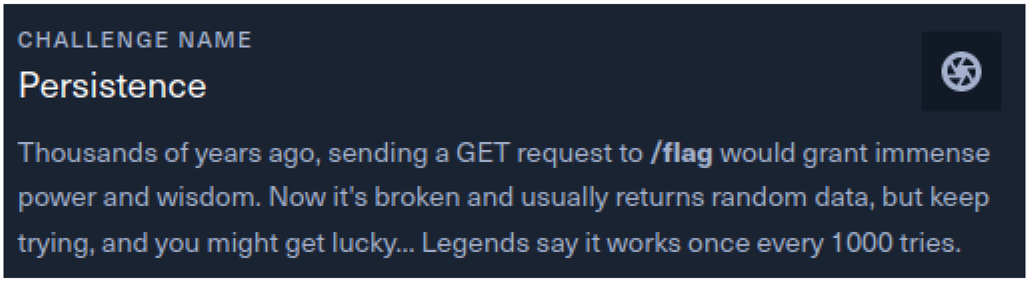
Challenge 3 – Persistence

Very Easy – Misc. Category

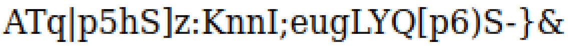


Persistence

Thousands of years ago, sending a GET request to **/flag** would grant immense power and wisdom. Now it's broken and usually returns random data, but keep trying, and you might get lucky... Legends say it works once every 1000 tries.

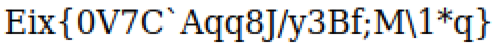
This being in the miscellaneous category we can be provided a number or different things for initial access. In this case we are given an IP address and a port number. From the description of the challenge, we will want to get a get request to the /flag directory.

When we do we are presented with a random string…



```
ATq|p5hS]z:KnnI;eugLYQ[p6)S-}&
```

We submit again and get a different string…



```
Eix{0V7C`Aqq8J/y3Bf;M\1*q}
```

Now we *can* keep refreshing until we hit the flag, but this would take a lot of time to do manually. In the description is says it works once every 1000 tries and I was not about to waste time hitting my F5 key.

So, we script…

```
import requests

url = "$IP"
search_string = "HTB{"
requests_per_check = 1000

count = 0
found = False

while not found:
    response = requests.get(url)
    if search_string in response.text:
        found = True
        print("Found search string in response!")
        print(response.text)
    count += 1
    if count % requests_per_check == 0:
        print(f"Checked {count} requests, still no match...")
    else:
        print(f"Checked {count} requests...", end="\r")
```

Ill break this script down…

First we want to import the Requests module, this will make sending an HTTP request very easy.

```
import requests
```

Then we want to set our values.

We set the URL we want to send the get requests to

We are looking for a specific response from the web server and in that response we know that the flag will start with “HTB{“ as seen in previous challenges.

And to make sure we don’t DOS the server because we were wrong about this being the solution we set the limit of 1000 requests.

```
url = "$IP"
search_string = "HTB{"
requests_per_check = 1000
```

Because we will need a loop in order to continue sending requests we set our count to 0, and because we do not need to keep requesting after we find the flag we set the found value.

```
count = 0
found = False
```

The next part of the code is the looping statement basically saying “ While not found do this…”

```
while not found:
```

Then we set the value for sending the get request by making use of the requests module and setting the argument to the URL value we set earlier.

```
response = requests.get(url)
```

Next the “If” statement is saying “if you find the string of “HTB{“ in the response change the value of “found” to True; ending the loop and printing out the response from the server.

```
if search_string in response.text:
        found = True
        print("Found search string in response!")
        print(response.text)
```

If the string is not found then it must add 1 to the counter

```
count += 1
```

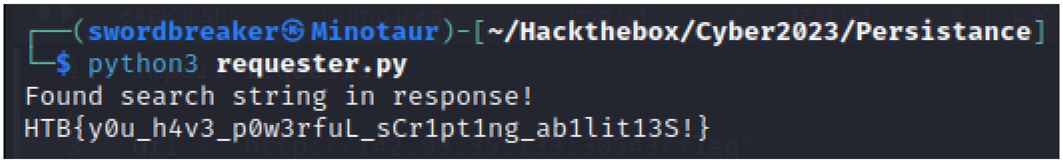
The next line is keeping track of the count value and placing it in a statement letting the user know how many requests the script has made.

```
if count % requests_per_check == 0:
    print(f"Checked {count} requests, still no match...")
```

Once the number of requests has surpassed the value set in the requests_per_check value it will move to the else statement and end the script.

```
else:
    print(f"Checked {count} requests...", end="\r")
```

Now to see if it works…



```
┌──(swordbreaker㉿Minotaur)-[~/Hackthebox/Cyber2023/Persistance]
└─$ python3 requester.py
Found search string in response!
HTB{y0u_h4v3_p0w3rfuL_sCr1pt1ng_ab1lit13S!}
```

Why thank you HTB!

Now the question of whether it would have been faster to spam F5 would have been faster then writing a script comes to mind. I know I certainly would have been upset if I pressed the F5 1 too many times and had to start over.