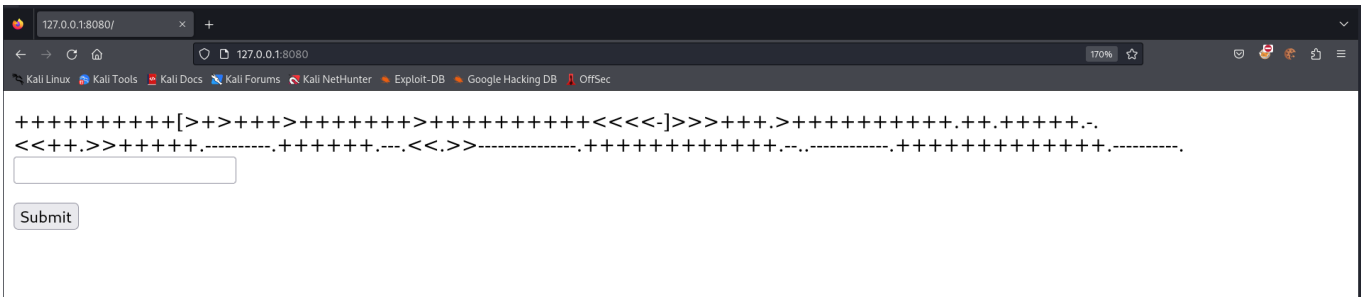
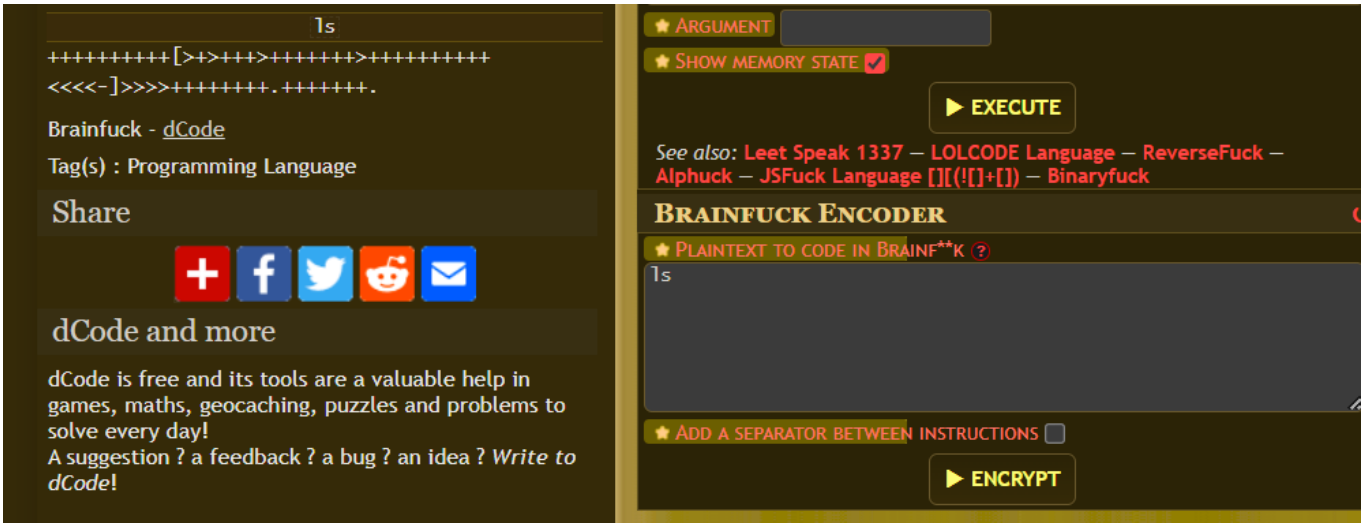
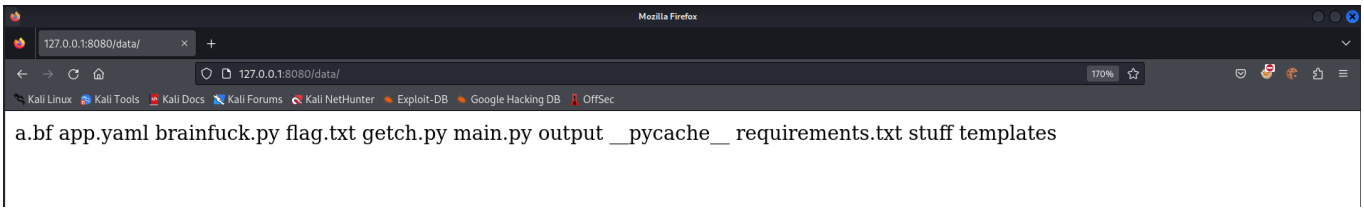
WTbF Writeup

Upon loading the webpage, the user is greeted with a piece of cipher text, followed by a submission form and a "Submit" button. The top piece of text is a piece of cipher text in "brainfuck" which, when decrypted, reads "Input your command".



After poking around, it is hoped that the user will realise they are executing commands on the web server and as such attempt to explore the current folder they are in which can be done through using a third party website to create payloads and then executing them. It is hoped that the user will uncover "flag.txt" in the directory they are currently exploring and download the file.!





Upon downloading and reading the file, there is an acronym for "GPG" which would be a hint for

the next phase of the challenge, which would be related to GPG keys.

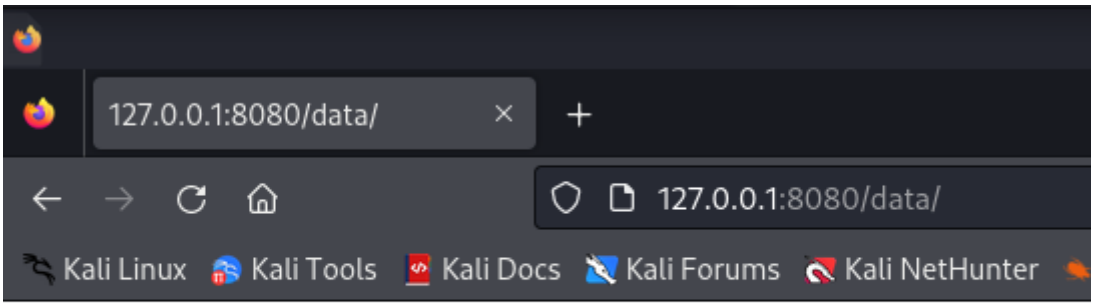
```
┌──(kali㉿kali)-[~]
└─$ wget http://127.0.0.1:8080/flag.txt
--2023-12-15 20:31:35--  http://127.0.0.1:8080/flag.txt
Connecting to 127.0.0.1:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22 [text/plain]
Saving to: 'flag.txt'

flag.txt            100%[===================>]      22  --.-KB/s    in 0s

2023-12-15 20:31:35 (2.51 MB/s) - 'flag.txt' saved [22/22]


┌──(kali㉿kali)-[~]
└─$ cat flag.txt
Good
Progress
Gandalf
```

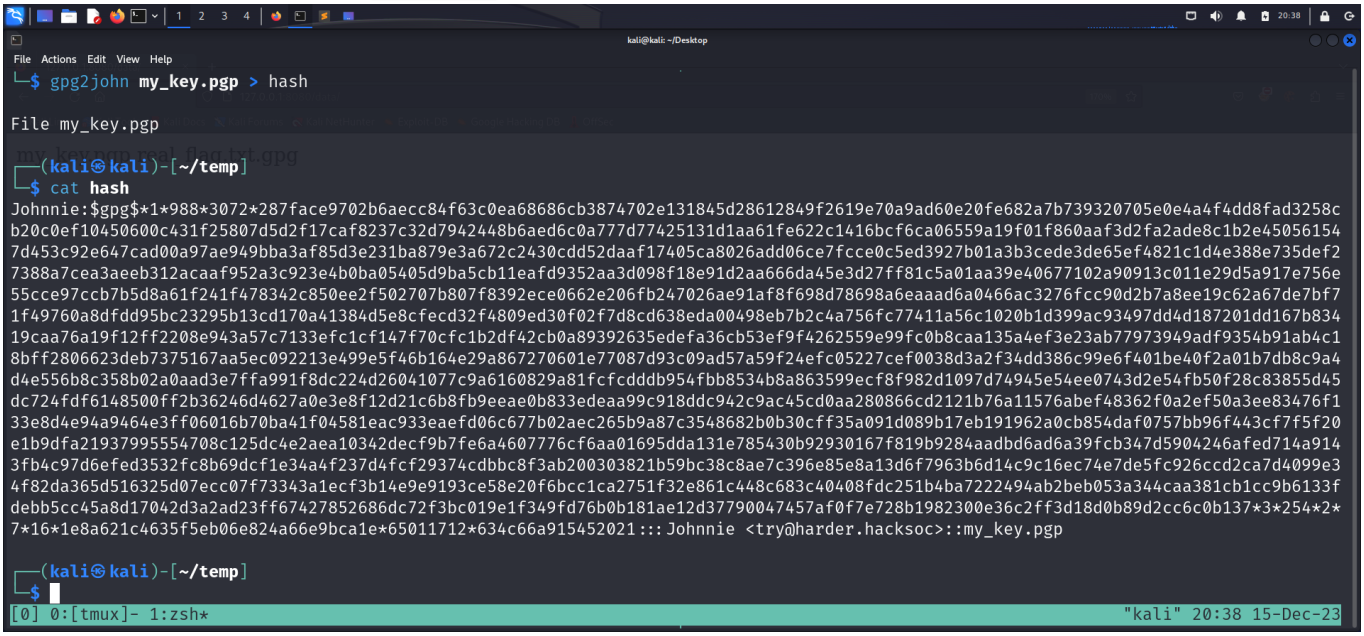
As noticed in an earlier screenshot, there is a folder called "stuff". Upon trying to view the files in the "stuff" folder the user would find two files, a PGP & GPG file.



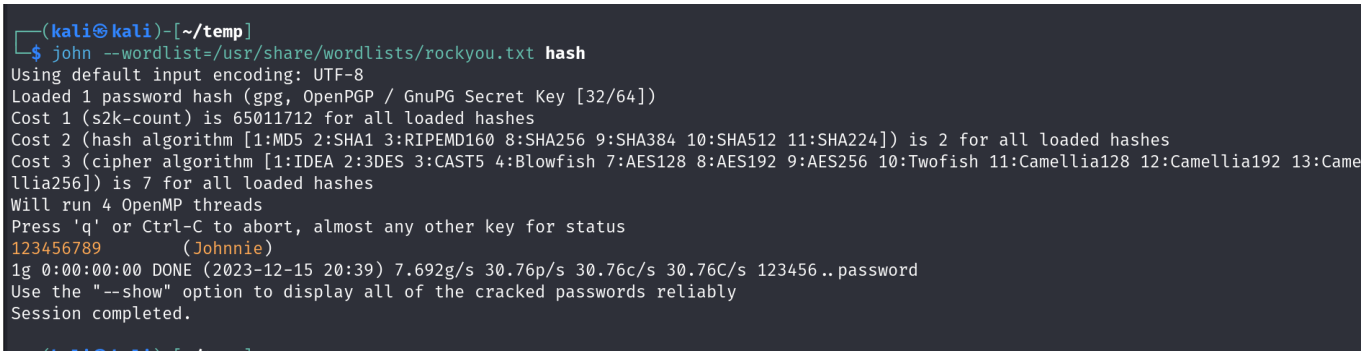
my_key.pgp real_flag.txt.gpg

The user should then download both files onto their machine. Due to the fact that we have a private key, and an encrypted message it is possible for the person to try and bruteforce the password of the private key. To start with, the user needs to convert `my_key.pgp` into a hash file,

this can be done through `gpg2john`



This "hash" file can now be passed into JohnTheRipper with the `rockyou.txt` wordlist and upon running this command, would output the password for the private key.



The user can then import the GPG key and decrypt the other file they found, revealing the real flag for this challenge.

```
┌──(kali㉿kali)-[~/temp]
└─$ gpg --import my_key.pgp
gpg: key 813D3C79ACF9C2BC: "Johnnie <try@harder.hacksoc>" not changed
gpg: key 813D3C79ACF9C2BC: secret key imported
gpg: Total number processed: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:  secret keys unchanged: 1
```

```
└─$ gpg --decrypt real_flag.txt.gpg > output
gpg: encrypted with 3072-bit RSA key, ID 2C9DAD3E3F9BFFD6, created 2023-12-15
      "Johnnie <try@harder.hacksoc>"

┌──(kali㉿kali)-[~/temp]
└─$ cat output
```