Pathological - Forensics

Category	Forensics
Solves	36
Points	186
CTF	CSCG 2022

Solution

The Challenge includes an `access.log` file, which contains information about incoming HTTP-requests, as well as the length of the response sent by the server.

I started by scrolling through the file and found something interesting starting at line **6335**. At this point some kind of brute-force for a password starts. I recognized the bruteforcing to be a binary search. Since you go to the next character after determining the current one, we have to just find the end of each characters brute-force session.

```
 /?type=\"]%20|%20//Password[substring(.,1,1)>\"%37\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,1,1)>\"%43\"]%20|%20//*[@Random=\"
1373
 /?type=\"]%20|%20//Password[substring(.,1,1)>\"%3d\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,1,1)>\"%40\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,1,1)>\"%41\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,1,1)>\"%42\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,2,1)>\"%4f\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,2,1)>\"%67\"]%20|%20//*[@Random=\"
1373
 /?type=\"]%20|%20//Password[substring(.,2,1)>\"%5b\"]%20|%20//*[@Random=\"
1373
 /?type=\"]%20|%20//Password[substring(.,2,1)>\"%55\"]%20|%20//*[@Random=\"
1373
 /?type=\"]%20|%20//Password[substring(.,2,1)>\"%52\"]%20|%20//*[@Random=\"
4184
 /?type=\"]%20|%20//Password[substring(.,2,1)>\"%53\"]%20|%20//*[@Random=\"
1373
 /?type=\"]%20|%20//Password[substring(.,3,1)>\"%4f\"]%20|%20//*[@Random=\"
1373
```

A shortened version of the data we have

We can see, that the character we are brute forcing switches from 1 to 2 [substring(.,1,1); ... substring(.,2,1);] after we checked for `0x42`, but we don't know, whether the actual char is one higher or lower than `0x42`. With a bit of experimentation we can assume, that a response length of **4284** instead of **1373** means, that we have to add one to the char we have received. I encapsulated the whole concept in this tiny script. *(**Note:** "sqlbrute" is a tidied-up version of the access.log file, which only contains the brute-forcing, with the index, character and response length seperated by pipes)*

```python
#!/usr/bin/env python3

with open("sqlbrute","r") as f:
    c = f.readlines()

last = "1"
flag = ""
last_char = "-"
last_up_down = "1373"


for i in c:
    index = i.split("|")[0].strip()
    char = i.split("|")[1].strip()
    up_down = i.split("|")[2].strip()

    if index != last:
        last = index
        if last_up_down == "4184":
            flag += chr(int(last_char,16)+1)
        else:
            flag += chr(int(last_char,16))


        print(flag)
    last_char = char
    last_up_down = up_down
```

The flag

The above script will magically give you the flag:

`CSCG{TempestGravitationLettuce}`