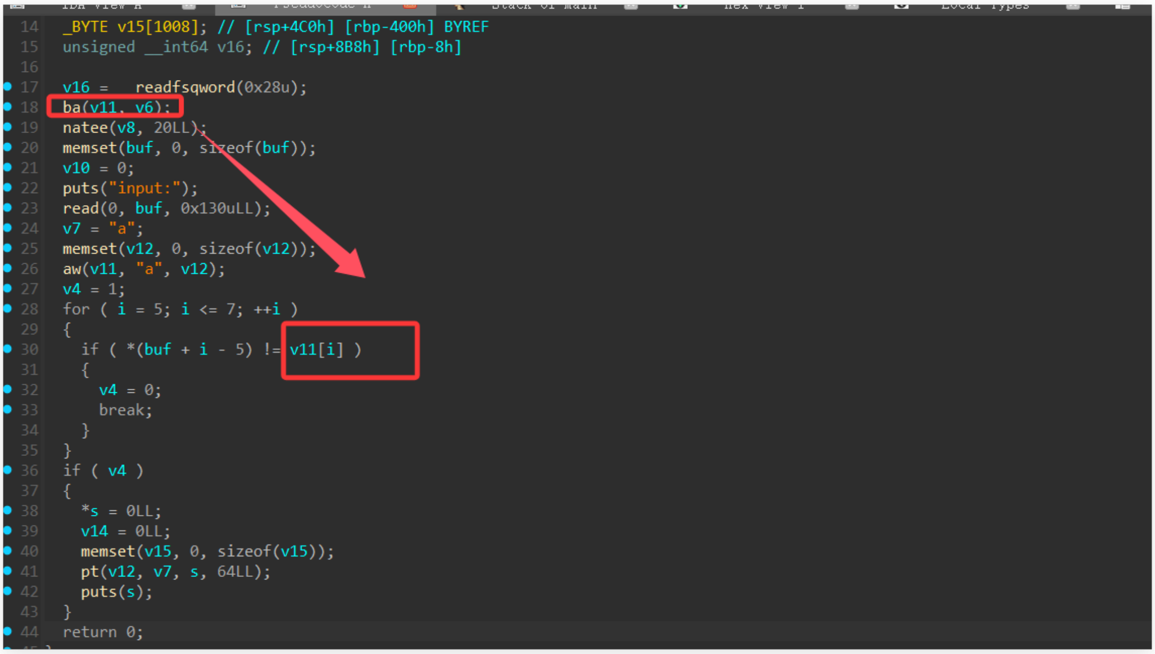
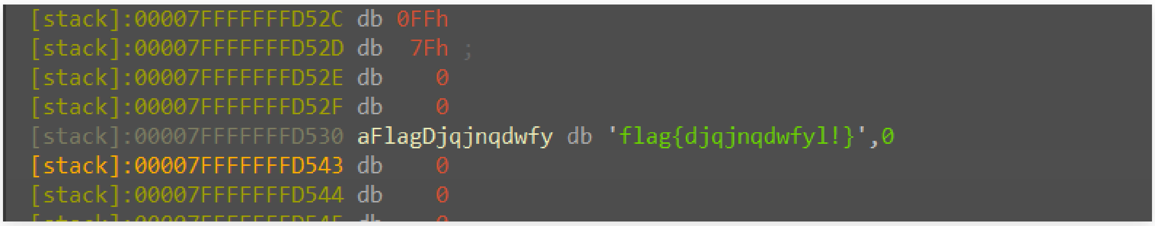
re

载入ida逻辑直接看了，v11用于input的check



```
_BYTE v15[1008]; // [rsp+4C0h] [rbp-400h] BYREF
unsigned __int64 v16; // [rsp+8B8h] [rbp-8h]

v16 = __readfsqword(0x28u);
ba(v11, v6);
natee(v8, 20LL);
memset(buf, 0, sizeof(buf));
v10 = 0;
puts("input:");
read(0, buf, 0x130uLL);
v7 = "a";
memset(v12, 0, sizeof(v12));
aw(v11, "a", v12);
v4 = 1;
for ( i = 5; i <= 7; ++i )
{
  if ( *(buf + i - 5) != v11[i] )
  {
    v4 = 0;
    break;
  }
}
if ( v4 )
{
  *s = 0LL;
  v14 = 0LL;
  memset(v15, 0, sizeof(v15));
  pt(v12, v7, s, 64LL);
  puts(s);
}
return 0;
```

那么直接动调跑起来看v11的值就好

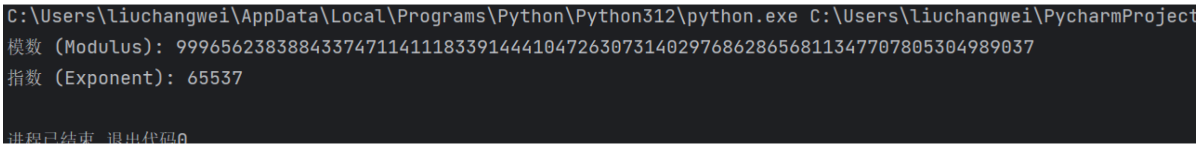


```
[stack]:00007FFFFFFFD52C db 0FFh
[stack]:00007FFFFFFFD52D db  7Fh ;
[stack]:00007FFFFFFFD52E db    0
[stack]:00007FFFFFFFD52F db    0
[stack]:00007FFFFFFFD530 aFlagDjqjnqdwfy db 'flag{djqjnqdwfyl!}',0
[stack]:00007FFFFFFFD543 db    0
[stack]:00007FFFFFFFD544 db    0
```

misc1

```
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.backends import default_backend

def load_rsa_public_key(pem_data):
    """从PEM格式的数据加载并返回RSA公钥对象。"""
    try:
        return serialization.load_pem_public_key(pem_data,
backend=default_backend())
    except Exception as e:
        print(f"加载公钥时发生错误: {e}")
        return None

def get_rsa_key_details(public_key):
    """提取并返回RSA公钥的模数和指数。"""
    if public_key:
        try:
            numbers = public_key.public_numbers()
            return numbers.n, numbers.e
        except Exception as e:
            print(f"提取公钥信息时发生错误: {e}")
    return None, None

def execute():
    public_key_pem = b"""
    -----BEGIN PUBLIC KEY-----
    MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhANOCibwA4MN7E2qRtAcdCjmFSflMuIX3
    Vrc/nzoUoaVtAgMBAAE=
    -----END PUBLIC KEY-----
    """
    public_key = load_rsa_public_key(public_key_pem)
    modulus, exponent = get_rsa_key_details(public_key)
    if modulus and exponent:
        print("模数 (Modulus):", modulus)
        print("指数 (Exponent):", exponent)

if __name__ == "__main__":
    execute()
```

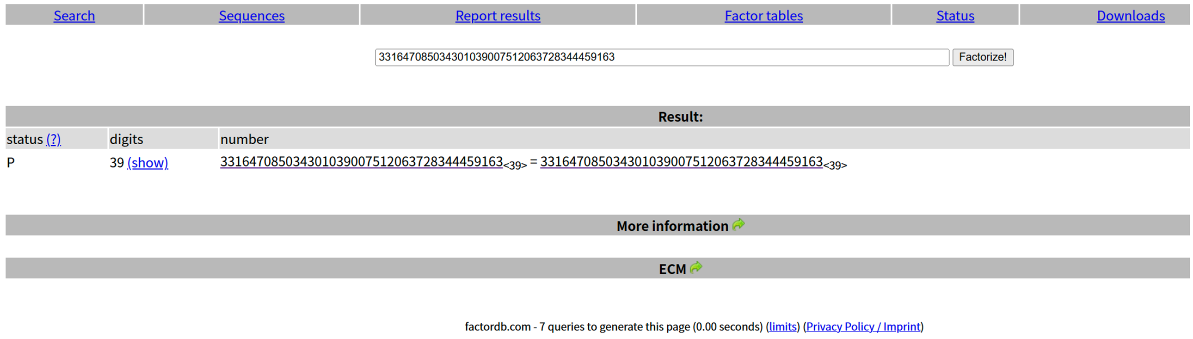
利用脚本提取公钥信息，模数和指数



```
C:\Users\liuchangwei\AppData\Local\Programs\Python\Python312\python.exe C:\Users\liuchangwei\PycharmProject
模数 (Modulus): 99965623838843374711411183391444104726307314029768628656811347707805304989037
指数 (Exponent): 65537

进程已结束，退出代码0
```

写脚本实现了扩展欧几里得算法（`extended_euclid`），用于计算最大公约数（GCD）和贝祖系数（`x` 和 `y`），并解决类似模逆的计算。使用扩展欧几里得算法计算一个数在模 `m` 下的逆元（`mod_inverse`），这是RSA算法中的一个重要步骤，用于计算私钥指数 `d`。

最后rsa解密



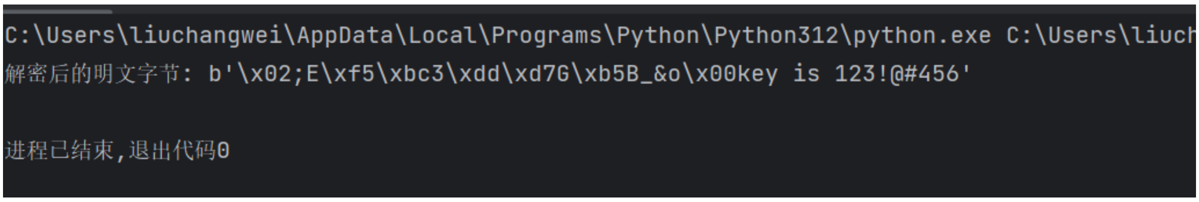
Result:

status (?)	digits	number
P	39 (show)	331647085034301039007512063728344459163<39> = 331647085034301039007512063728344459163<39>

factordb.com - 7 queries to generate this page (0.00 seconds) (limits) (Privacy Policy / Imprint)

```
def extended_euclid(a, b):
    """
    扩展欧几里得算法，计算最大公约数和贝祖系数
    返回最大公约数、x 和 y，使得 ax + by = gcd(a, b)
    """
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = extended_euclid(b % a, a)
        return (g, x - (b // a) * y, y)

def compute_mod_inverse(a, m):
    """
    计算模逆元
    使用扩展欧几里得算法计算 a 在模 m 下的逆元
    如果逆元不存在，抛出 ValueError 异常
    """
    g, x, _ = extended_euclid(a, m)
    if g != 1:
        raise ValueError('模逆元不存在')
    return x % m

def rsa_decrypt_with_primes(ciphertext, p, q, e):
    """
    使用 p, q, e 进行 RSA 解密
    :param ciphertext: 密文整数
    :param p: 质数 p
    :param q: 质数 q
    :param e: 公钥指数
    :return: 解密后的明文整数
    """
    try:
        # 计算 n 和 ϕ(n)
        n = p * q
        phi_n = (p - 1) * (q - 1)

        # 计算私钥指数 d
        d = compute_mod_inverse(e, phi_n)

        # 解密：m = c^d mod n
        plaintext = pow(ciphertext, d, n)
        return plaintext
    except ValueError as ve:
        print(f"解密时发生错误: {ve}")
        return None

def perform_decryption():
    # 示例参数
    p = 301421686937198008750983790559102741399  # 质数
    q = 331647085034301039007512063728344459163  # 质数
    e = 65537  # 公钥指数

    try:
        # 从文件中读取密文
        with open("venus.en", "rb") as f:
            encrypted_data = f.read()

        # 将字节数据转换为整数
        ciphertext = int.from_bytes(encrypted_data, byteorder="big")

        # 解密
        plaintext = rsa_decrypt_with_primes(ciphertext, p, q, e)

        if plaintext is not None:
            # 将明文整数转换回字节数据
            plaintext_bytes = plaintext.to_bytes((plaintext.bit_length() + 7) //
8, byteorder="big")
            print("解密后的明文字节:", plaintext_bytes)
    except FileNotFoundError:
        print("未找到 venus.en 文件，请检查文件路径。")
    except Exception as e:
        print(f"发生未知错误: {e}")

if __name__ == "__main__":
    perform_decryption()
```

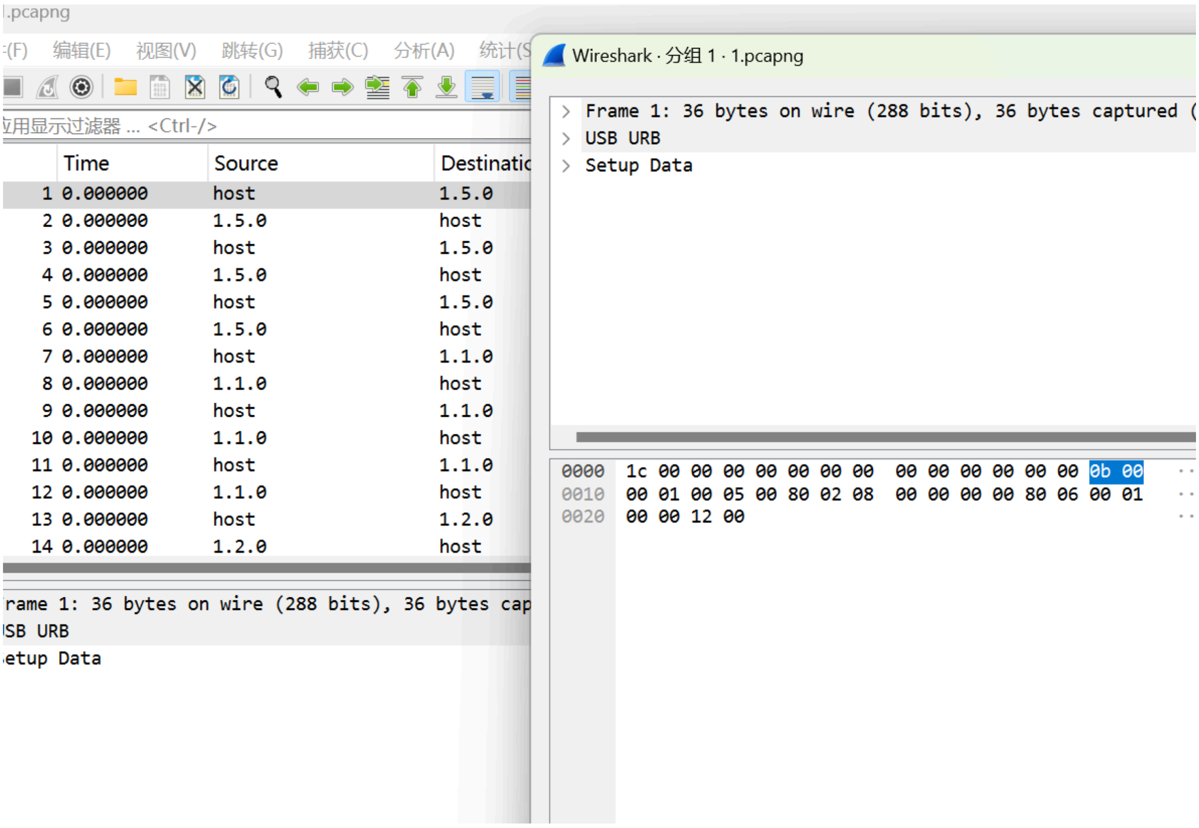
得到密钥123!@#456，解压缩得到flag



```
C:\Users\liuchangwei\AppData\Local\Programs\Python\Python312\python.exe C:\Users\liuch
解密后的明文字节: b'\x02;E\xf5\xbc3\xdd\xd7G\xb5B_&o\x00key is 123!@#456'

进程已结束,退出代码0
```

misc2

010破解伪加密，wireshark分析为键鼠USB流量



.pcapng
:(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S
立用显示过滤器 ... <Ctrl-/>
Wireshark · 分组 1 · 1.pcapng

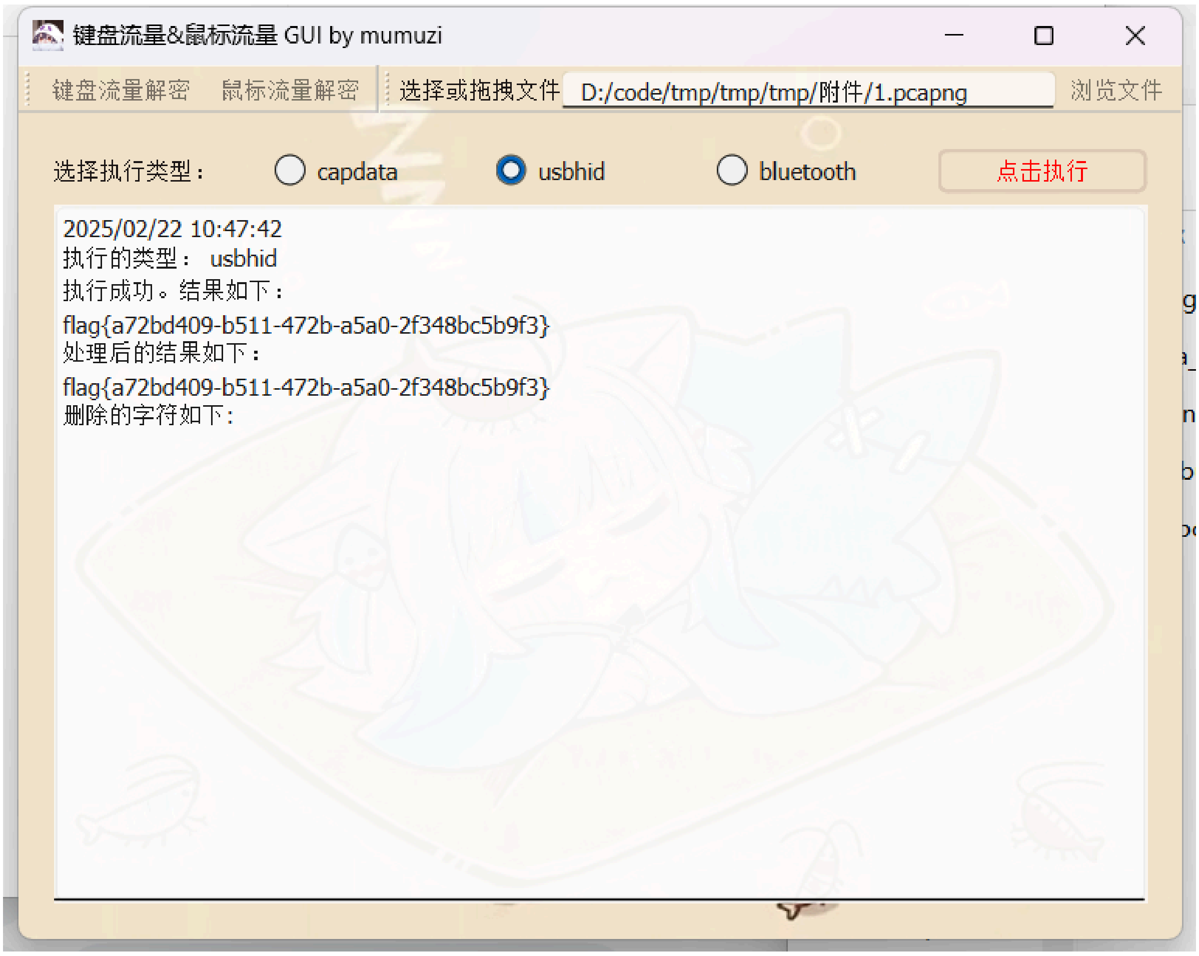
```
   Time       Source    Destinatic
 1 0.000000   host      1.5.0
 2 0.000000   1.5.0     host
 3 0.000000   host      1.5.0
 4 0.000000   1.5.0     host
 5 0.000000   host      1.5.0
 6 0.000000   1.5.0     host
 7 0.000000   host      1.1.0
 8 0.000000   1.1.0     host
 9 0.000000   host      1.1.0
10 0.000000   1.1.0     host
11 0.000000   host      1.1.0
12 0.000000   1.1.0     host
13 0.000000   host      1.2.0
14 0.000000   1.2.0     host
```

```
rame 1: 36 bytes on wire (288 bits), 36 bytes cap
ISB URB
etup Data
```

```
> Frame 1: 36 bytes on wire (288 bits), 36 bytes captured (
> USB URB
> Setup Data
```

```
0000  1c 00 00 00 00 00 00 00  00 00 00 00 00 00 0b 00   ..
0010  00 01 00 05 00 80 02 08  00 00 00 00 80 06 00 01   ..
0020  00 00 12 00                                        ..
```

工具直接梭哈，flag大写



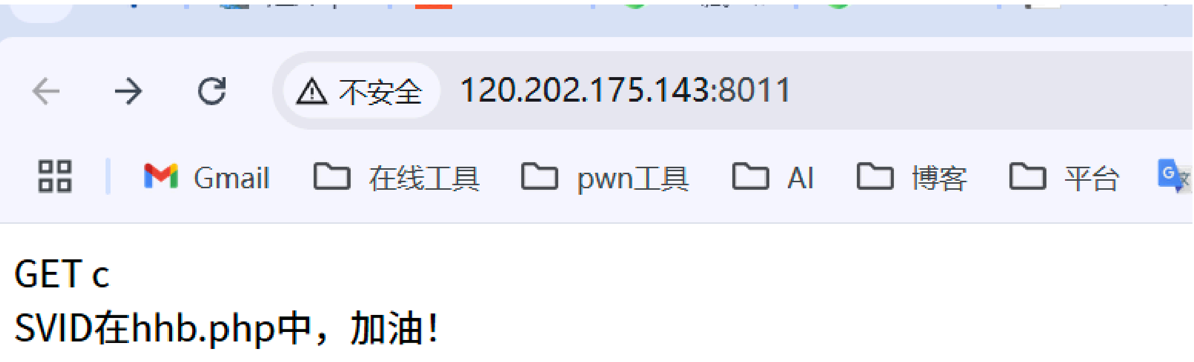
选择或拖拽文件 D:/code/tmp/tmp/tmp/附件/1.pcapng

选择执行类型： ○ capdata ● usbhid ○ bluetooth

```
2025/02/22 10:47:42
执行的类型： usbhid
执行成功。结果如下：
flag{a72bd409-b511-472b-a5a0-2f348bc5b9f3}
处理后的结果如下：
flag{a72bd409-b511-472b-a5a0-2f348bc5b9f3}
删除的字符如下：
```

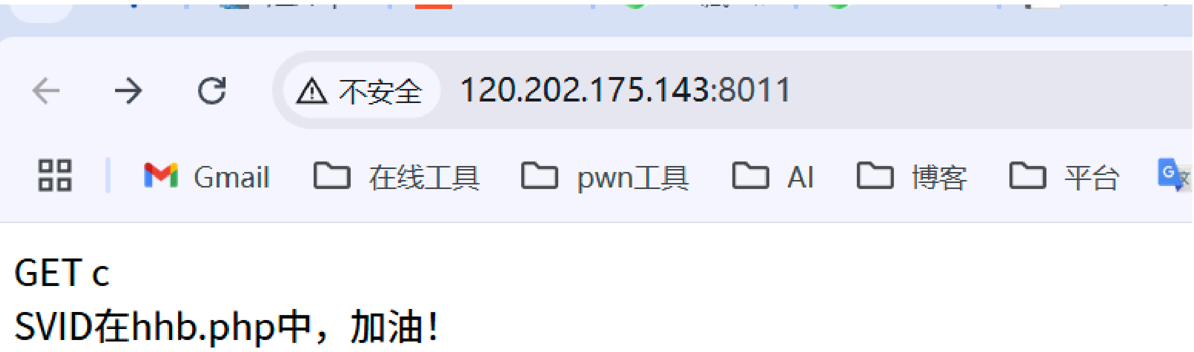
web1

根据提示，传参c成功得到密文

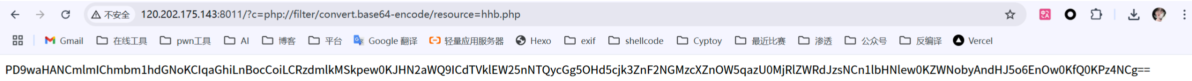


120.202.175.143:8011

GET c
SVID在hhb.php中，加油！

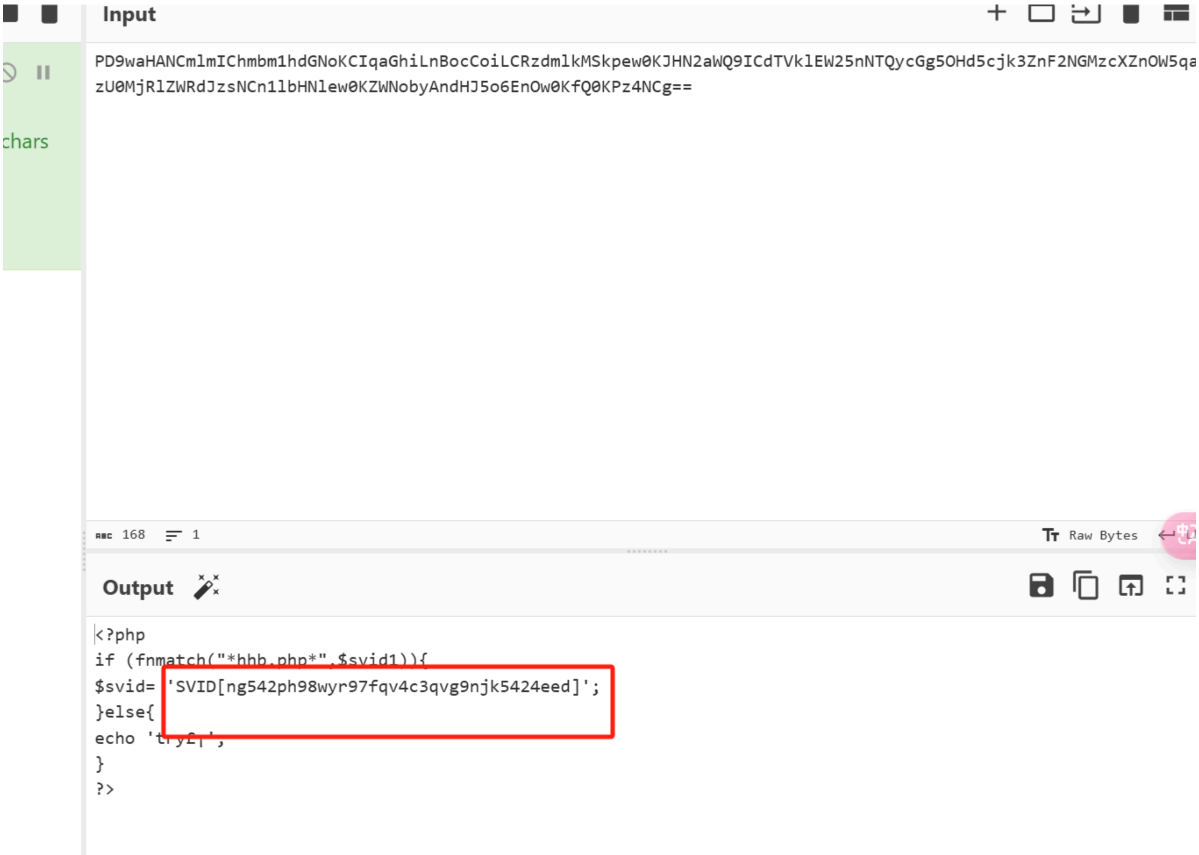


GET c
SVID在hhb.php中，加油！



PD9waHANCmlmIChmbm1hdGNoKCIqaGhiLnBocCoiLCRzdmlkMSkpew0KJHN2aWQ9ICdTVklEW25nNTQycGg5OHd5cjk3ZnF2NGMzcXZnOW5qazU0MjRlZWRdJzsNCn1lbHNlew0KZWNobyAndHJ5o6EnOw0KfQ0KPz4NCg==

base64解密得到flag



PD9waHANCmlmIChmbm1hdGNoKCIqaGhiLnBocCoiLCRzdmlkMSkpew0KJHN2aWQ9ICdTVklEW25nNTQycGg5OHd5cjk3ZnF2NGMzcXZnOW5qa
zU0MjRlZWRdJzsNCn1lbHNlew0KZWNobyAndHJ5o6EnOw0KfQ0KPz4NCg==

```
<?php
if (fnmatch("*hhb.php*",$svid1)){
$svid= 'SVID[ng542ph98wyr97fqv4c3qvg9njk5424eed]';
}else{
echo 'try2!';
}
?>
```

web2

万能的sql密码

admin/admin成功得到flag