

2016 年京津冀大学生网络安全知识技能挑战赛 writeup

队名: C-cUp

成员: 杨昌鑫、李南均、王程

0x01 签到题

扫一扫，反正我是没扫出来，队友说是用百度扫的



0x02 Hello

查看页面的代码发现如下表达式，要求 `sha1(var)==Ciphertext`，显然

这里的 Ciphertext 应该是个变量

```
1  </head>
2
3  <body>
4  <!--ROUND ONE-->
5  <!--找出明文 -->
6  <!--sha1(三个数字+fF8JpdJFFgEXSXS1TXuherA7B0iMR)==Ciphertext -->
7  <!--请于10s内提交答案 -->
8  <div class="panel">
9  <div class="wrap">
10 <form method="POST" action="#">
```

发现请求页面时响应头部有 Ciphertext 的值。



那思路就很清晰了：访问页面获得 sha1() 括号中的字符串以及头部的 Ciphertext 的值，通过暴力穷举的方式找到满足条件的三个数字脚本如下

```
10 full_url='http://106.75.67.214:2250/?pass=0debc27caa70ce48#'
11 ycx = requests.session()
12 data = ycx.post(full_url)
13 Data = data.content
14 head = data.headers["Ciphertext"]
15
16 string_list = Data.split()
17 result = string_list[19]
18 result = result[22:52]
19
20
21 final = list(itertools.permutations([0,1,2,3,4,5,6,7,8,9],3))
22 for i in range(0,720):
23     my_best = ''.join('%s' % id for id in final[i])
24     print my_best+result
25     secret = hashlib.sha1(my_best+result).hexdigest()
26     #print secret
27     if(hashlib.sha1(my_best+result).hexdigest()==head):
```

跑了之后竟然得到这个，还需要算个表达式；

```
974kwGN7KexeA5kco4U7FAEv9LgqVQDPg
974
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hello</title>
  <link rel="stylesheet" href="css/style.css" media="screen" type="text/css">
</head>

<body>
<!--ROUND TWO-->
<!--找出明文-->
<!--算出值提交：499*533*3+1-1*4*297-4*516-14+4+282+5-427-91-48+55-->
<!--请于10s内提交答案-->
  <div class="panel">
    <div class="wrap">
      <form method="POST" action="#">
        <input type="text" name="pass" placeholder=" here"/>
        <button onclick="form.submit();">Shhh!</button>
      </form>
    </div>
  </div>
<div style="text-align:center;clear:both">
```

大同小异，继续提取页面中的表达式 eval() 计算提交即可：

```

499*533*3+1-1*4*297-4*516-14+4+282+5-427-91-48+55
794416
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hello</title>
  <link rel="stylesheet" href="css/style.css" media="screen" type="text/css">
</head>

<body>
<!--找出明文 -->
<!--flag{f325c62b-9505-4c13-ad4b-010bddb23c68} -->
<!--请于10s内提交答案 -->
  <div class="panel">
    <div class="wrap">
      <form method="POST" action="#">
        <input type="text" name="pass" placeholder=" here"/>
        <button onclick="form.submit();">Shhh!</button>
      </form>
    </div>
  </div>
<div style="text-align:center;clear:both">
</div>

```

0x03 crack MD5

访问页面得到如下返回，大意是第二行的是 md5 密文，其对应的明文就是第三行的那些字母，不过顺序被打乱了。

Brute force crack MD5

Let's play a game. Please crack the MD5 below, which is generated by

0e417e85dfe6767dbeda423449e9d614

[1U,Ir1Nh

看明白题思路就清晰了：首先访问页面，提取密文及明文字符串，对明文字符串进行排列组合并加密，将结果与已给的密文比较即可

```

27
28 final = list(itertools.permutations([result[0],result[1],result[2],result[3],result[4],result[5],result[6],result[7],result[8]],9))
29
30 #print len(final)
31
32 for i in range(0,362880):
33     my_best = ''.join(final[i])
34     if(md5(my_best)==secret):
35         print my_best
36         success_url = "http://106.75.67.214:2050/?code="+my_best
37         flag = ycx.get(success_url)
38         print flag.content
39         break
40     else:

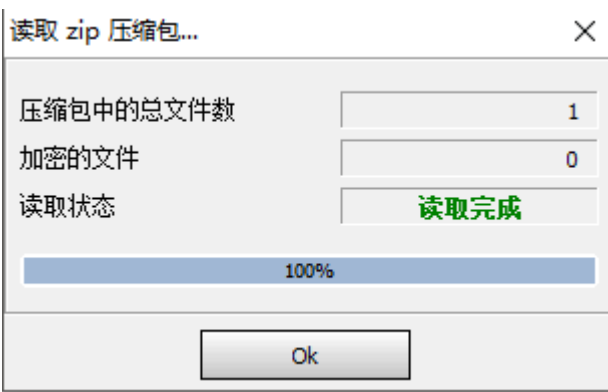
```

但是这样做很耗时，大部分情况都是 too late，不过还好侥幸跑出来一次。

```
try_again
Z6[zPd3P(
<!DOCTYPE html>
<html>
<head>
    <title>Brute force crack MD5</title>
</head>
<body>
<h1>Brute force crack MD5</h1>
flag{25e54f71-6bfe-4962-811c-454aa9a07e62}</body>
</html>
```

0x04 破解

文件是个压缩包，解压还要密码，丢到某破解软件中，提示加密的文件为 0。



那应该是伪加密了，用十六进制编辑器打开，将文件相应的标志位 0900 修改为 0800 即可。

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	00	4B	03	04	14	00	00	00	08	00	0B	7D	2F	49	80	74	PK.....}/I t
00000016	87	CB	F3	8F	1D	00	DD	1B	26	00	0A	00	00	00	6E	6F	嘛?..?&....no
00000032	66	6C	61	67	2E	67	69	66	CC	9A	E7	53	13	DC	F3	F6	flag.gif虎鏢.?藥
00000048	D3	49	08	25	40	80	84	1A	7A	87	D0	41	8A	09	1D	A4	親.%@e? z?豎S..?
00000064	F7	26	86	2A	4D	A4	89	20	A8	49	E8	BD	23	20	60	E8	? ? M?? \杞#.:?
00000080	5D	E9	4D	C5	50	A5	89	54	45	40	A5	A9	80	0D	B0	A1]?M?P?途E@才?啊
00000096	B7	86	E7	FB	2F	9E	33	D7	CC	99	D9	7D	75	3E	B3		槽現? ? 滋機)u>?
00000112	D7	EE	79	B1	A6	E6	26	5A	DA	DE	DF	81	58	E0	2F	00	最y?~&Z譜? X?/.
00000128	00	06	60	61	02	B0	21	00	5C	CC	00	34	12	C0	CD	0A	.. a.?!\?.4.???
00000144	C0	B2	02	F8	D9	01	82	28	80	30	37	40	96	07	20	C7	啦.?? ? @?@? ?
00000160	0B	50	C4	00	94	F9	01	1A	02	00	2D	41	C0	39	1C	E0	.P? 旗...-A?..?
00000176	3C	0E	40	10	05	18	8A	03	4C	24	80	16	92	C0	0B	D2	<.@...? L\$@捌?.?
00000192	40	6B	59	A0	BD	2C	D0	41	0E	E8	24	0F	74	56	04	BA	@kY?? 豎.?\$.tV.?
00000208	2B	01	2F	E2	81	5E	CA	40	92	0A	D0	5B	15	E4	A7	0A	+. /?. 豎? 卩坏.??
00000224	F2	57	FB	3F	05	A8	83	02	35	40	41	1A	A0	60	2D	50	馬非?..?? 5@A. 燻-P
00000240	98	36	E8	AA	0E	28	4A	17	7C	4D	0F	1C	A3	0F	8E	D5	? 璿.(J. M..? 幷
00000256	07	DF	20	80	13	88	EO	DB	06	60	B2	21	84	6A	04	49	. ? €. 20藥..? 百l. T

解压文件得到 noflag.gif。



并没有什么，想用记事本打开图片，竟然把我电脑卡死了。继续用 winhex 打开。

3 45 78 61 52 7A 42 33 56 44 4A	WVlExcExaKzB3VDJ
3 48 5A 54 52 56 55 30 56 56 52	3NFFWcHZTRVUOVVR
A 7A 42 57 56 7A 6C 52 56 46 56	oblRuZzBWVz1RVFV-
5 6A 5A 55 61 30 35 46 55 31 68	aRWRGUjZUa05FU1hE
7 6B 39 6B 57 47 52 58 54 6A 4A	GMU0xWk9kWGRXTjJ7
2 58 5A 57 56 54 6C 4A 59 55 56	kRGR6RXZWVT1JYUV
A 45 35 5A 65 6B 35 32 55 56 64	RM1ZYZE5Zek52UVd
A 48 46 6A 56 55 70 4E 56 46 56	aTlMyZHFjVUpNVFVv
F 54 42 61 53 47 68 5A 54 6C 5A	WV2NXOTBaSGhZT1ZE
1 6D 39 53 52 45 4A 31 57 6C 52	kTlRuQm9SREJ1W1R
4 58 4A 6A 4D 6C 5A 61 5A 46 4D	oSmJqTXJjM1ZaZFM

发现有一大长串似乎是 base64 码的字母，提取出来解码，发现解码后还是 base64，有情况啊！继续解码，还是 base64，三次之后解出如下字符串，很明显是一副 gif。

用 base64 直接解码成图片

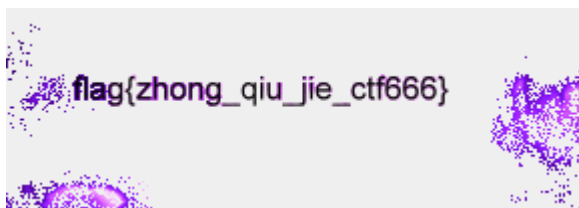


以下是您的 Base64 代码所解码出来的图片，右键另存为保存图片。



返回

还是动的，用 gif 编辑器打开得到 flag



0x05 tryhard

加密部分算法，flag{*****}每个字符会生成一个两位的 16 进制形式的字符，而密文 40 位，所以 flag{*****}共 20 位，其中 key=*****长 14 位，对于从 flag{*****}开始的每一个字符，加密过程由 key 中的[0,7],[1,8].....位依次参与迭代运算：

```
for c in flag:
    print c
    c = ord(c)
    print c
    for a, b in zip(key[0:n], key[n:2*n]):
        print a, b
        c = (ord(a) * c + ord(b)) % 251
    print c
    encrypted += '%02x' % c
    print encrypted
print encrypted
print len(encrypted)
```

加密算法迭代化简一下就更清晰了：

$$\begin{aligned}C_1 &= [a_1 \cdot c_0 + b_1] \% 251 \\C_2 &= [a_2 \cdot C_1 + b_2] \% 251 \\&\vdots \\C_7 &= [a_7 \cdot C_6 + b_7] \% 251\end{aligned}$$

代入得：

$$\begin{aligned}C_2 &= (a_2 \cdot a_1 \cdot c_0 + a_2 \cdot b_1 + b_2) \% 251 \\C_3 &= (a_3 \cdot a_2 \cdot a_1 \cdot c_0 + a_3 \cdot a_2 \cdot b_1 + a_3 \cdot b_2 + b_3) \% 251 \\&\vdots \\C_7 &= (a_7 \cdot a_6 \cdot a_5 \cdots a_1 \cdot c_0 + a_7 \cdot a_6 \cdots a_2 \cdot b_1 + \cdots + b_7) \% 251\end{aligned}$$

其中 $(a_7 \cdot a_6 \cdot a_5 \cdots a_1)$ 和 $(a_7 \cdot a_6 \cdots a_2 \cdot b_1 + \cdots + b_7)$ 对于 $flag\{*****\}$ 每一位的加密运算都一样，因此不妨将其当做两个整体 i 与 j

$$\therefore C_j = i \cdot c_0 + j \% 251$$

其中 C_j 即代表 $flag\{*****\}$ 每一个经加密后的值（转16进制即可）

由于 $flag\{*****\}$ 中 $flag\}$ 及其对应的密文是已知的，在 $\text{mod}251$ 之后 i, j 可认为是 $[0, 251]$ 之间的数，带进去暴力破解即可

```
1 import sys
2 if sys.argv[4] == 0x23 and data[5] == 0x41:
3     rev_key = 'de3d93ed23a293b4dec3f0b4b4deb1d2a5e1c341'
4     test = []
5     result = []
6     for i in range(0, 5):
7         test.append(rev_key[2*i:i*2+2])
8         result.append(int(test[i], 16))
9     print result
10
11 flag = 'flag{'
12
13 for i in range(0, 251):
14     for j in range(0, 251):
15         secret = []
16         for k in range(0, 5):
17             secret.append((i * ord(flag[k]) + j) % 251)
18         if secret == result:
19             print i, j
```

求得 i, j 分别等于 15 和 198。

```
root@kali:~/Desktop/tryhard# python reverse.py
[222, 61, 147, 237, 35]
15 198
```

现已知 i, j 那么密文中 $flag\{*****\}$ 对应 $*****$ 的那一串根据化简出来的表达式，反带即可


```

rev_key = 'de3d93ed23a293b4dec3f0b4b4deb1d2a5e1c341'
test = []
result = []
for i in range(0,20):
    test.append(rev_key[2*i:i*2+2])
    result.append(int(test[i],16))
print result
print len(result)

key = []
for i in range(0,14):
    for j in range(0,251):
        if((15*j+198)%251==result[i+5]):
            key.append(chr(j))
print key

```

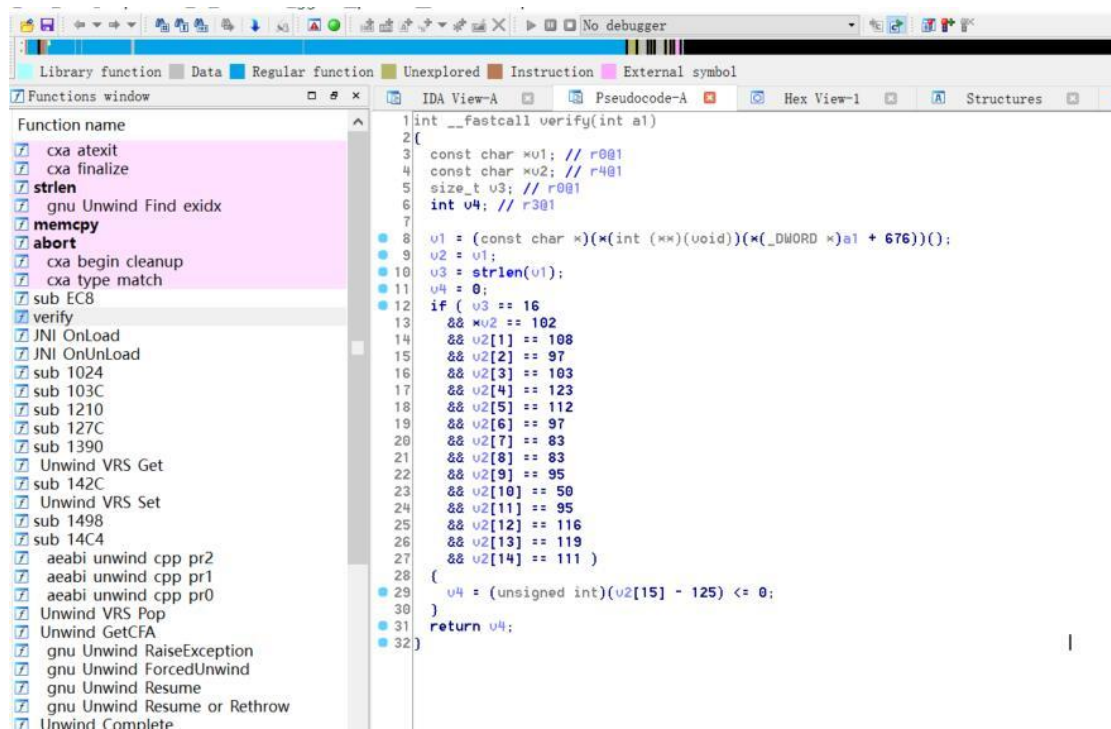
```

[222, 61, 147, 237, 35, 162, 147, 180, 222, 195, 240, 180, 180, 222, 177,
65, 225, 195, 65]
20
['b', 'a', '1', 'f', '2', '5', '1', '1', 'f', 'c', '3', '0', '4', '2']

```

0x06 re50

这是我队友做的，打完比赛他就出去玩了，我帮他写一下，不周之处还请包涵。IDA 载入看到 verify 函数，里面的 v2[] 数组即是 flag



0x07 cry50

首先 base64 解码密文 Ot7lAO72opsedkxTngbD3FhwP50x8sosA f9oL

OkIpr8PN7J0OmQ7nWxvgvaiRn+Tp95zcTDj,

```
for n in range(salt_length):
    salt += chr(random.randrange(256))
data = salt + crypt(data, sha1(key + salt).digest())
if encode:
    data = encode(data)
return data
```

解码后的密文前 16 位即是 salt，后面是 crypt()加密的结果，key 已知，salt 已知，则 sha1(key+salt).digest()可知，然后将结果及解码后密文 crypt()加密的部分丢进去，解密即可得到 flag

```
a2YsP9zCpXUd01+CaemC5oCVQNQpPzXCbSWKZDYZnKZfXXn7/r+GEBuzseKa0w==
flag{197a9b67f0ebc04647c1eeefcec99808}
root@kali:~/Desktop/cry100#
```