

# 数据库2021“安恒·泰山杯”山东省网络安全大赛测试赛部分题目 write up

转载

[knight11112](#) 于 2021-12-28 09:48:40 发布 76 收藏

文章标签: [数据库](#) [web安全](#) [database](#)

原文链接: [www.awaedu.com](http://www.awaedu.com)

版权

## 数据库2021“安恒·泰山杯”山东省网络安全大赛测试赛部分题目 write up

### 2021“安恒·泰山杯”山东省网络安全大赛测试赛部分题目 wp

- [web1 adminlogin](#)
- [web2 ezphp](#)
- [misc1 extractall](#)
- [misc2 认真你就输了](#)
- [crypto1 rsa17](#)
- [crypto2 移位凯撒](#)
- [reverse ezgo](#)
- [pwn ez\\_rop](#)

线上测试赛，两个多小时8道题做出来5道，感觉自己还是太菜，简单记录一下。能down下来的东西都在最后百度云链接里。

### web1 adminlogin

## 扫描得到admin.php

御剑WEB目录扫描优化版

扫描域名:  请求方式:  线程:  超时:

主要状态码  
 200  3xx  403 其他状态码:  [状态码格式](#)  自动保存扫描结果 [查看](#)

ID	字典名称	行数
<input checked="" type="checkbox"/>	1 ASP	1854
<input checked="" type="checkbox"/>	2 ASPX	822
<input checked="" type="checkbox"/>	3 DIR	1153
<input checked="" type="checkbox"/>	4 JSP	631
<input checked="" type="checkbox"/>	5 MDB	419
<input checked="" type="checkbox"/>	6 PHP	1066

```
进度: 100%  
耗时: 00:00:24.1473623  
结果: 3  
任务结束...
```

ID	URL	Code
1	http://183.129.189.60:10045/admin.php	200
2	http://183.129.189.60:10045/index.php	200
3	http://183.129.189.60:10045/config.php	200

CSDN @失控的菜鸡玩家



得到登录页面，根据提示flag在数据库里，推测是sql注入。

试了一下substr、database等等能过滤的都过滤了，有点烦，想留到最后做结果也没来得及做。

## web2 ezphp

# Where is flag?

CSDN @失控的菜鸡玩家

源代码:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Welcome</title>
</head>
<body >
</div>
<h1 style="text-align: center">Where is flag?</h1>
<!--
foreach ($_POST as $item => $value){
  $$item=$value;
  $secret = $$item;
}
foreach ($_GET as $key => $value){
  if ($key=='flag'){
    $str=$value;
    $$str=$secret;
  }
}
if (isset($hehe)){
  echo "
```

".\$hehe."

"; } //flag+flaag=DASCTF{XXXXXXXX} --> </body> <center> </html>

The screenshot shows a web browser window displaying the PHP Manual page for the `foreach` loop. The browser's address bar shows `? for` and the page title is `break?`. The manual content includes the following text:

**foreach**

(PHP 4, PHP 5, PHP 7)

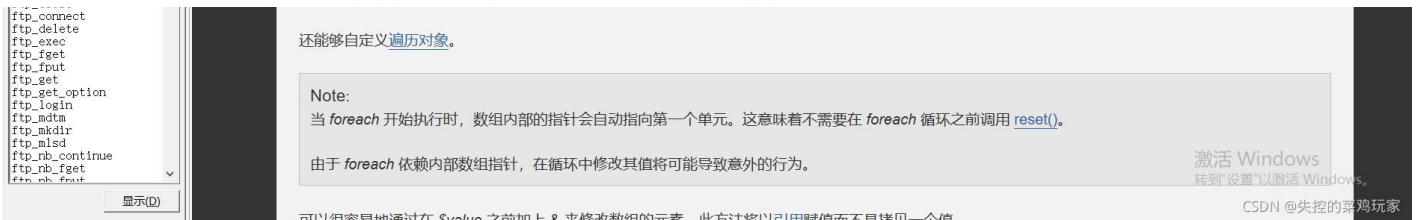
*foreach* 语法结构提供了遍历数组的简单方式。*foreach* 仅能够应用于数组和对象，如果尝试应用于其他数据类型的变量，或者未初始化的变量将发出错误信息。有两种语法：

```
foreach (array_expression as $value)
    statement

foreach (array_expression as $key => $value)
    statement
```

第一种格式遍历给定的 *array\_expression* 数组。每次循环中，当前单元的值被赋给 *\$value* 并且数组内部的指针向前移一步（因此下一次循环中将会得到下一个单元）。

第二种格式做同样的事，只除了当前单元的键名也会在每次循环中被赋给变量 *\$key*。

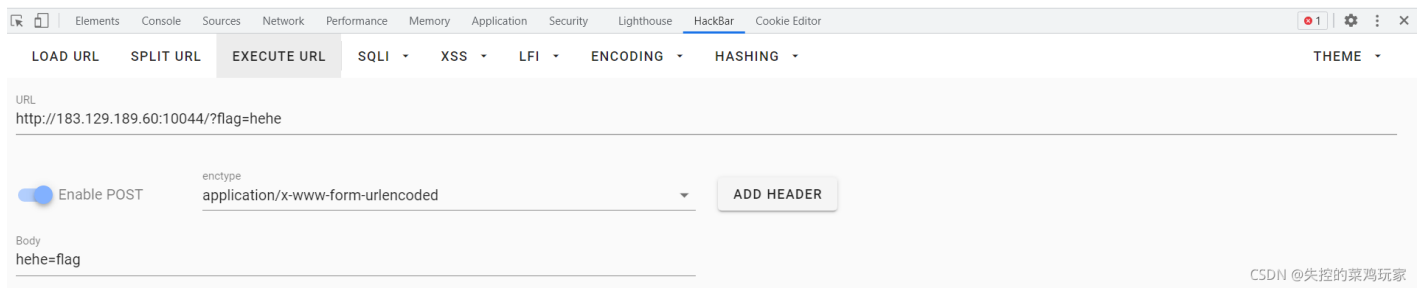


代码审计一下，是一个变量覆盖漏洞，具体漏洞原理可参考其他大佬的博客：

<https://www.cnblogs.com/zjzjdbk/p/12985530.html>

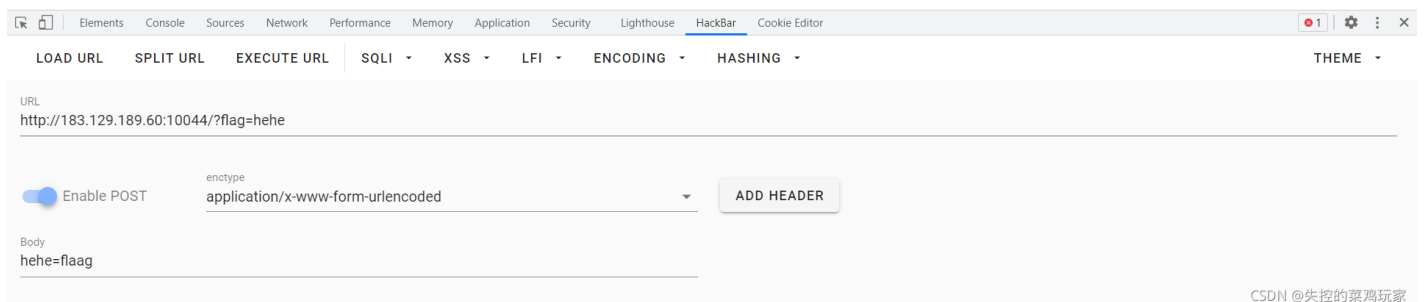
## Where is flag?

DASCTF{27b62da69



## Where is flag?

e01bf1ad3e4e737c2b8f4a1}



将两部分拼一下就能得到flag。

<http://www.zuowenge.cn>

wp来自另外一位大神：

先写一个多层zip解压脚本，并且注意到zip文件名似乎是有规律的，这里顺便把文件名也记录了一下。

```

import zipfile
import os

name = '[REFTQ]'

s = []
while True:
    s.append(name)
    with zipfile.ZipFile(name + '.zip', 'r') as ziip:
        ziip.extractall(pwd=name.encode())
        delname = name
        name = ziip.filelist[0].filename[0:6]

    print(s)
    os.remove(delname + '.zip')

```

最深处有个图片，调整图片宽高，最下面有个提示who am i。



who am i

CSDN @失控的菜鸡玩家

有点熟悉，百度一下是斐波那契，想到斐波那契数列，再结合文件名的特点，写出下面的脚本。

```

s = ['[REFTQ]', '1RGe0V', '4dHJhY', 'eht8on', '3RhbGx', '8smjqt', 'zmckit', 'fSXNfU', '9rskp5', 'a93su6', 'a

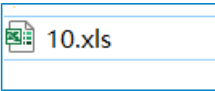
print(len(s))
fbnachi = list(map(int, '1、2、3、5、8、13、21'.split(',')))
res = ''
for i in fbnachi:
    res += s[i-1]
print(res)

```

base64解密一下即得flag

```
DASCTF{Extractall_Is_So_Fun}
```

misc2 认真你就输了



附件是一个Excel文件，但是打开提示文件已经损坏，放到kali里看一下格式。

```
(root@kali)~[~/Documents]
# ls
10.xls  debug  dvcs-ripper-master  easyre.exe  F5-steganography-master  GitHack-master  linux_server64  output  re

(root@kali)~[~/Documents]
# file 10.xls
10.xls: Microsoft Excel 2007+

(root@kali)~[~/Documents]
# binwalk 10.xls
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Zip archive data, at least v2.0 to extract, compressed size: 19, uncompressed size: 17, name: xl/charts/flag.txt
67	0x43	Zip archive data, at least v1.0 to extract, name: docProps/
106	0x6A	Zip archive data, at least v2.0 to extract, compressed size: 404, uncompressed size: 872, name: docProps/app.xml
556	0x22C	Zip archive data, at least v2.0 to extract, compressed size: 322, uncompressed size: 608, name: docProps/core.xml
925	0x39D	Zip archive data, at least v1.0 to extract, name: xl/
958	0x3BE	Zip archive data, at least v2.0 to extract, compressed size: 1441, uncompressed size: 9895, name: xl/calcChain.xml
2445	0x98D	Zip archive data, at least v1.0 to extract, name: xl/charts/
2485	0x985	Zip archive data, at least v2.0 to extract, compressed size: 1064, uncompressed size: 3398, name: xl/charts/chart1.xml
3599	0xE0F	Zip archive data, at least v2.0 to extract, compressed size: 996, uncompressed size: 3001, name: xl/charts/chart2.xml
4645	0x1225	Zip archive data, at least v1.0 to extract, name: xl/drawings/
4687	0x124F	Zip archive data, at least v2.0 to extract, compressed size: 397, uncompressed size: 1013, name: xl/drawings/drawing1.xml
5138	0x1412	Zip archive data, at least v2.0 to extract, compressed size: 397, uncompressed size: 1014, name: xl/drawings/drawing2.xml
5589	0x15D5	Zip archive data, at least v1.0 to extract, name: xl/drawings/_rels/
5637	0x1605	Zip archive data, at least v2.0 to extract, compressed size: 179, uncompressed size: 293, name: xl/drawings/_rels/drawing1.xml.rels
5881	0x16F9	Zip archive data, at least v2.0 to extract, compressed size: 179, uncompressed size: 293, name: xl/drawings/_rels/drawing2.xml.rels
6125	0x17ED	Zip archive data, at least v2.0 to extract, compressed size: 272, uncompressed size: 613, name: xl/sharedStrings.xml
6447	0x192F	Zip archive data, at least v2.0 to extract, compressed size: 598, uncompressed size: 1377, name: xl/styles.xml
7088	0x1BB0	Zip archive data, at least v1.0 to extract, name: xl/theme/
7127	0x1BD7	Zip archive data, at least v2.0 to extract, compressed size: 1467, uncompressed size: 7079, name: xl/theme/theme1.xml
8643	0x21C3	Zip archive data, at least v2.0 to extract, compressed size: 366, uncompressed size: 687, name: xl/workbook.xml
9054	0x235E	Zip archive data, at least v1.0 to extract, name: xl/worksheets/
9098	0x238A	Zip archive data, at least v2.0 to extract, compressed size: 838, uncompressed size: 2706, name: xl/worksheets/sheet1.xml
9990	0x2706	Zip archive data, at least v2.0 to extract, compressed size: 1881, uncompressed size: 12007, name: xl/worksheets/sheet2.xml
11925	0x2E95	Zip archive data, at least v2.0 to extract, compressed size: 1127, uncompressed size: 6779, name: xl/worksheets/sheet3.xml
13106	0x3332	Zip archive data, at least v2.0 to extract, compressed size: 1520, uncompressed size: 10201, name: xl/worksheets/sheet4.xml

binwalk命令看到一大堆压缩包，将文件后缀改为zip，挨个翻一下文件得到flag.txt。

此电脑 > 本地磁盘 (C:) > ctf > 比赛文件 > 0929测试赛 > 认真你就输了的附件 > 10 > 10 > xl > charts

名称	修改日期	类型	大小
chart1.xml	1980/1/1 16:00	XML 文档	4 KB
chart2.xml	1980/1/1 16:00	XML 文档	3 KB
flag.txt	2016/8/11 19:14	文本文档	1 KB

## crypto1 rsa17

给了一段代码：



```

from Crypto.Util.number import *
import binascii
import gmpy2
flag = '*****'
hex_flag=int(flag.encode("hex"),16)

p=getPrime(256)
q=getPrime(256)
n=p*q

e=0x3
c1=pow(hex_flag,e,n)
c2=pow(hex_flag+1,e,n)

print("n=",hex(n))
print("e=",hex(e))
print("c1=",hex(c1))
print("c2=",hex(c2))

...
('n=', '0xb28ae8f29f8b90e8b8c5667b2b71e49929446b41f7f7a3e9e45bc52a1e8c45d59c1788be48a9c365d51feee0b2cd32950')
('e=', '0x3')
('c1=', '0x7ba5502ecbc3b15ad8c2db8f30a593eb062dde4d7dfacadf0a28291d1a576389a18dfba0607c0243f843f637449089dd')
('c2=', '0x891ac4f663df41c1f6433ee3513d749c3ba02fe0aacd7f51d791b9bac4f7e5194bd484d78d972c344faf600f7d3aa580')
...

```

e=3, 想到关联信息攻击

## 2. 关联信息攻击 - $e=3$ , $m[2] = a*m[1] + b$

写一个脚本:

```

#!/usr/bin/python
# -*- coding: utf-8 -*-
import gmpy2

n = 9351035609579912430580224362406913775216485260866801060250235841497131649675821473038044490729550589638
e = 3
c1 = 6475853636479050645596496086080582816789963066323389815672770714308619633711541909793891052802547415987
c2 = 7180748878269451580223627474056868509561249251375351465737365567454518806786657134253453800373864987758

def get_m1(a, b, c1, c2, n):
    a3 = pow(a, 3, n)
    b3 = pow(b, 3, n)
    tmp1 = ((c2 + 2*a3*c1 - b3) * b) % n
    tmp2 = ((c2 - a3*c1 + 2*b3) * a) % n
    tmp3 = gmpy2.invert(tmp2, n)
    tmp4 = (tmp1 * tmp3) % n
    return tmp4

m1 = get_m1(1, 1, c1, c2, n)
print(m1)
#m1=0b218774971804085528558358969417603446702511773250904882108354386662296629999644383681650889865769009

```

将十进制转化为字符串得到flag:

```
C:\Users\XuanJian>python3
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> bytes.fromhex(hex(2187749718040855285583589694176034467025117732509048821083543866622966299996443
83681650889865769009)[2:])
b'flag is :3e7f54b8ad38787670776c2698a67c01'
```

## crypto2 移位凯撒

密文: ch\at;X[hUeQZcNU\_QL^f

c的ascii码+3=f的ascii码值

h的ascii码+4=l的ascii码值

\的ascii码+5=a的ascii码值

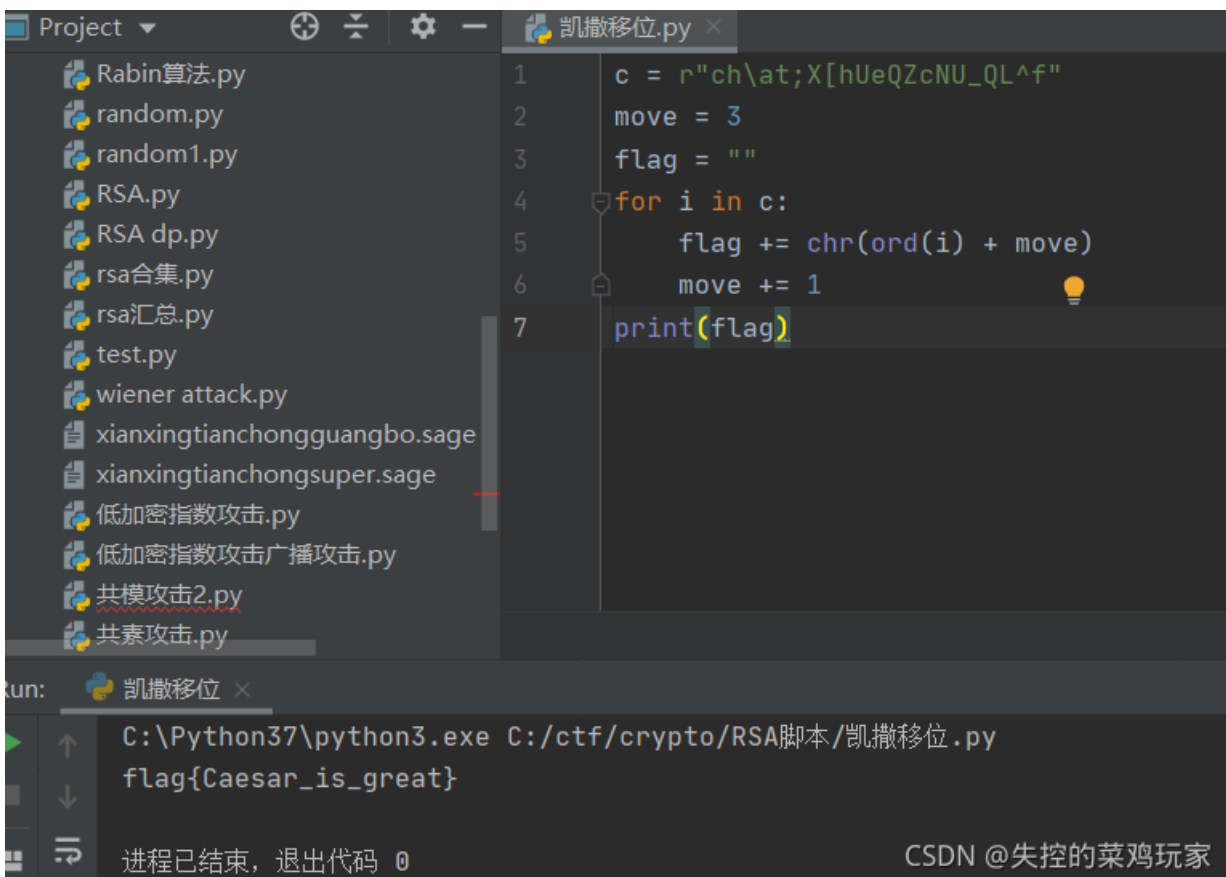
a的ascii码+6=g的ascii码值

t的ascii码+7={的ascii码值

前面拼起来就是flag{

所以是一个变形的凯撒密码，密文和明文的差值从3开始递增

```
c = r"ch\at;X[hUeQZcNU_QL^f" / #这里一定要加r, 不然\会被当作转义符
move = 3
flag = ""
for i in c:
    flag += chr(ord(i) + move)
    move += 1
print(flag)
```



## reverse ezgo



## 赛题详情

🕒 本题用时: 135分18秒

题目名称: ezGo

题目内容: GoGoGo

题目分值: 80.0

题目难度: 困难

相关附件: ezGo的附件.zip

下载

CSDN @失控的菜鸡玩家

看标题，go语言的逆向，本菜鸡暂时只学过c语言的逆向，时间有限，比赛中暂时搁置了，文件保存了，以后做出来补充。

## pwn ez\_rop

🕒 本题用时: 146分34秒

题目名称: easy\_rop

题目内容: I think this is the only way for pwner. Let's take a challenge. 靶机地址 183.129.189.60:10046

题目分值: 30.0

题目难度: 容易

相关附件: easy\_rop的附件

下载

CSDN @失控的菜鸡玩家

时间有限，比赛中暂时搁置了，文件保存了，以后做出来补充。

除了web1都能down下来，百度云链接：链接：<https://pan.baidu.com/s/19bdOz9z5i3CHQLqC3P7P4g>

提取码: mipu

—来自百度网盘超级会员V5的分享