[xueqi]ISCC 2019 writeup 信息安全与对抗-解题思路xueqi

转载

h2cf 于 2019-10-12 09:10:22 发布
分类专栏： CTF 文章标签： ctf iscc 2019 安全竞赛
原文链接：https://www.idxueqi.cn/ctf/103.html

ISCC 2019 writeup-全国大学生信息安全与对抗技术竞赛 解题思路xueqi

WEB

Web1

题目地址: http://39.100.83.188:8001

```
<?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__);
```

读了下代码，get传参，value和password。Value的值要等w3lc0me_To_ISCC2019，但ascii值不能在32和127之间。

查php手册资料得知chr（）函数是 值除以256取余数。这就好办了，只需要在原有数值上加上256就ok了。

再看password ，intval可以用科学技术法绕过，如：2e4

```
http://39.100.83.188:8001/index.php? value[]=375&value[]=307&value[]=364&value[]=355&value[]=304&value[]=36
```

Hello w3lc0me_To_ISCC2019!

flag{[redacted]}

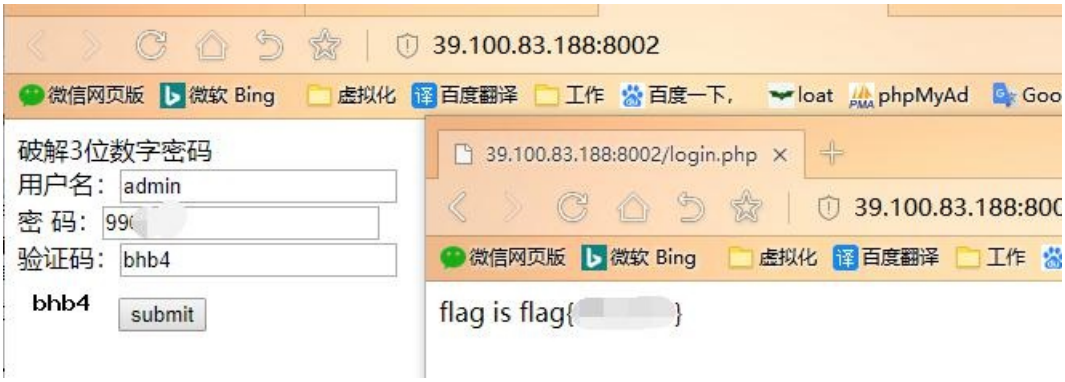
```php
<?php
error_reporting(0);
require 'flag.php';
$value = $_GET['value'];
$password = $_GET['password'];
$username = '';

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__);
```

Web2

此题是要输入三位数值，还有验证码，要暴力破解试下。

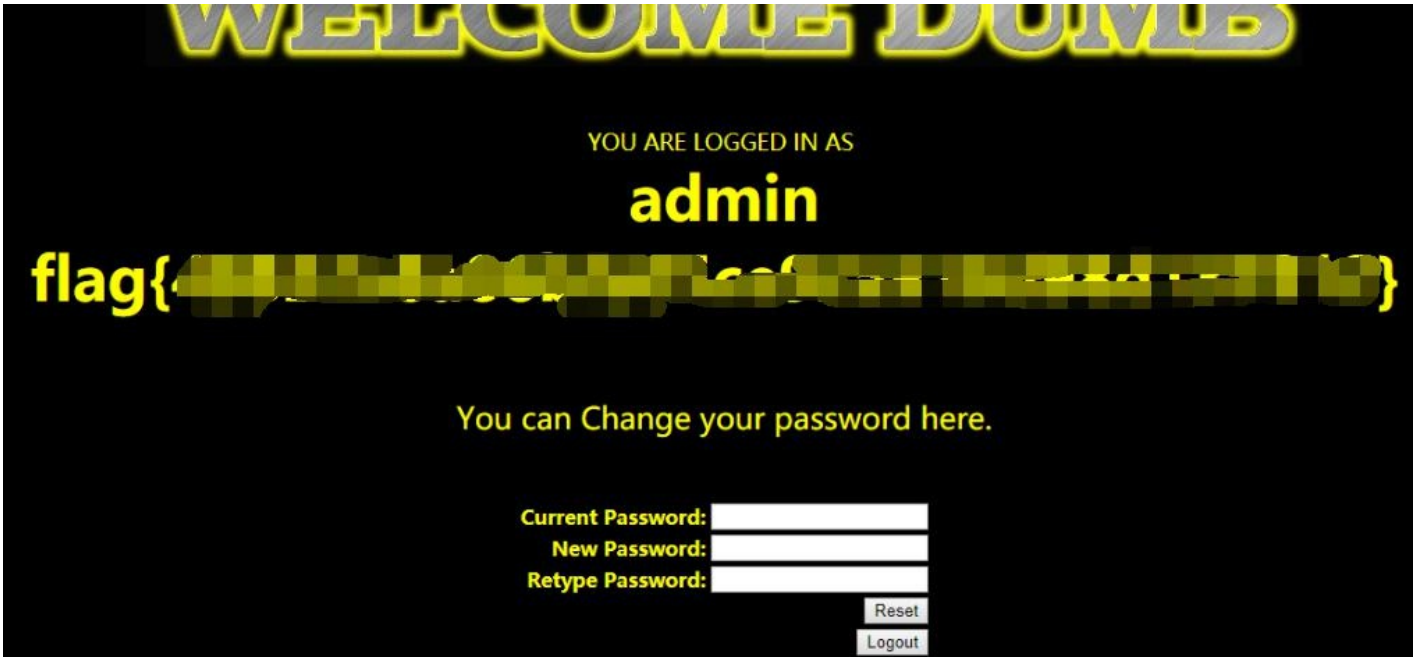
自己从999倒着试试，还没到990就试出来了

不过大佬告诉我要抓包，删除cookie和user_code的值，填上pwd的值就好了



Web3

sqli-labs中的一道原题，好像加了过滤。自己做题时，admin好像被大佬改了弱口令，捡漏登陆admin直接爆出flag，后来被修复。
思路是sql注入，注册帐号登陆后，修改密码页面，帐号填admin'or 1# 修改admin的密码，然后登录admin获取flag



Web4

http://39.100.83.188:8066

```
<?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__);
}?>
```

读下代码，parse_url这个函数引起注意，百度了下，原来存在变量覆盖漏洞。

Key要=0 ，key是被sha256加密过的，传一个hashed_key，将原来的密文覆盖掉，hashed_key的值是0经过sha256加密的密文

构造

```
?action=auth&hashed_key=5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9&key=0
```

Web5

访问，提示不是Union.373组织成员。看到这个就想到改请求头。
User-agent中加上Union.373，再次访问，然后页面提示要用户名，post方式提交，username和password的值
试试admin访问，提示密码既为flag。问了大佬，说要在这里注入，有过滤，union没过滤，

Web6

一开始无从下手，顺着PasteBin查阅原来是考验JWT。注册一个帐号，随便鼓捣一番，有list功能呢，抓包发现jwt到https://jwt.io/ 解码查看。
查看源码发现common.js文件有大料：
function getpubkey(){

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

把jwt更改加密方式为HS256用公钥加密
然后拿到"admin:22f1e0aa7a31422ad63480aa27711277"
访问url/text/admin:22f1e0aa7a31422ad63480aa27711277获得flag

----------2019-5-20更新----------

MISC

Misc1 隐藏的信息

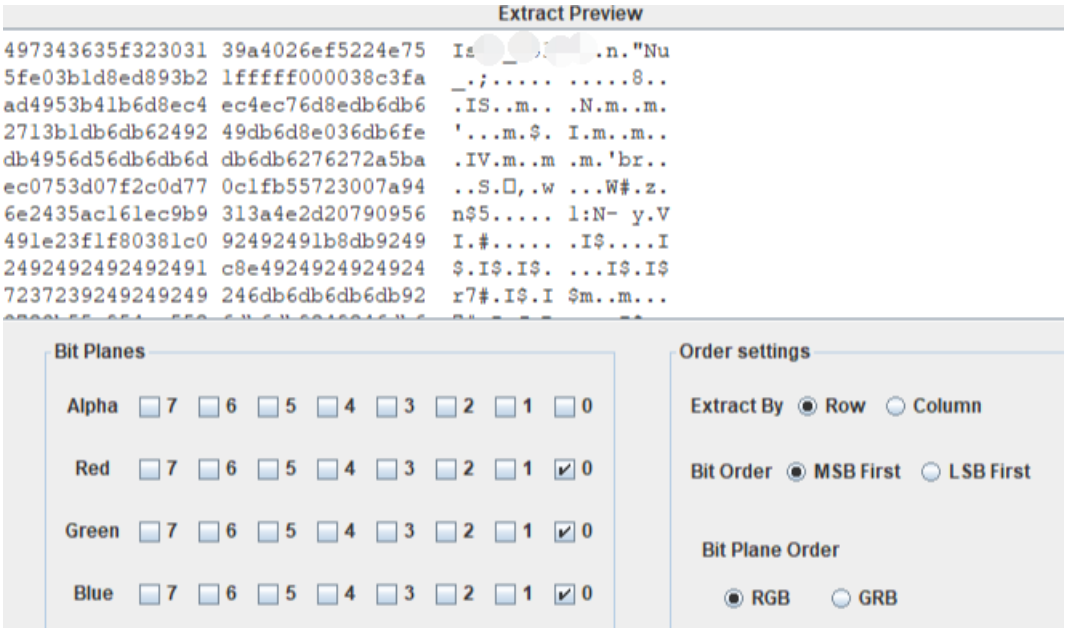
这是一个被混淆的文件，但是我忘记了这个文件的密码。你能够帮助我还原明文吗？

> 0126 062 0126 0163 0142 0103 0102 0153 0142 062 065 0154 0111 0121
> 0157 0113 0111 0105 0132 0163 0131 0127 0143 066 0111 0105 0154 0124
> 0121 060 0116 067 0124 0152 0102 0146 0115 0107 065 0154 0130 062 0116
> 0150 0142 0154 071 0172 0144 0104 0102 0167 0130 063 0153 0167 0144
> 0130 060 0113

看这些数字，像是某种进制，最大是7，可能是8进制，转换十进制再对应ascii表转字符串。得到结果。

Misc 倒立屋

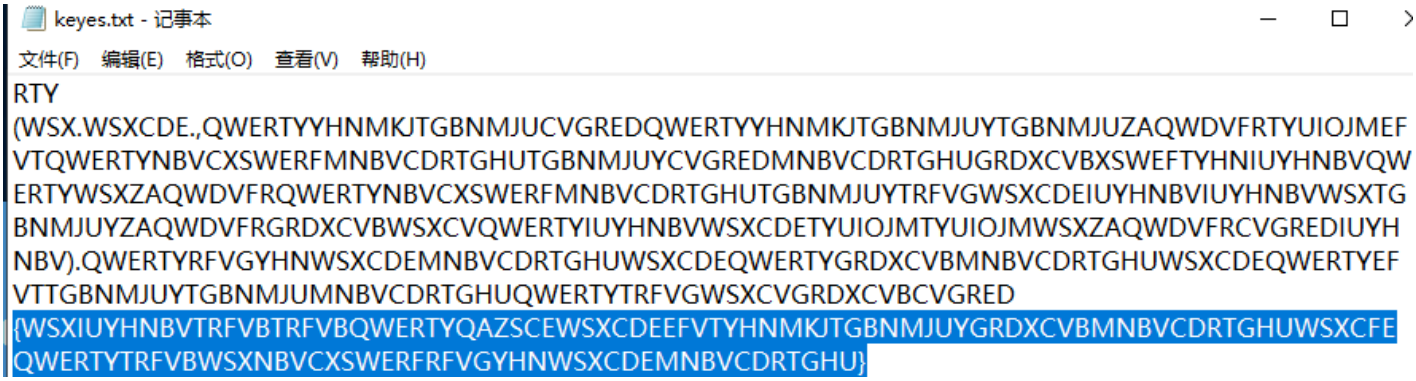
拿到压缩包，里面是张图，还以为线索在倒立上，用Stegsolve打开，发现是lsb隐写。



选择 R G B 最低为勾选，就可看到iscc字段，不过flag提交有个脑洞就是要把找到的线索反序排列提交。

Misc Keyes' secret

打开时，看到一大串字符 头都大了，也不知道是什么加密什么算法。睡一觉起来，发现有一段{}括住的字符串。



解密无果，后来试了下键盘轨迹，发现轨迹前四个是iscc，此题需要有想象力，字母对照键盘画轨迹。

Misc Aesop's secret

打开是个gif图片，快速闪动，能看出来是个图案，拖进pscs6里，所有图层勾选可视。发现iscc图形旁边是黑点，还以为线索在周围黑色像素点上，拖进hex编辑器发现gif尾部存在一串加密密文，怀疑是base64，复制解密，无果。

拿去试AES解密，解不出，密钥不知道，想起gif的iscc字样，密码填入iscc，解密拿到明文。

Misc Welcome-流浪

这个题目拖了五天才有思路，期间大佬们一直调戏我这萌新，说电影里有线索，想到出题人的脑洞，我还真去看了遍电影，专去找户口身上的付款二维码，条码，刘培强的号码牌条码（气到痛哭）。
然后有大佬提示我留意“户口”“长条”，然后就和小伙伴展开脑洞，长条是李一一，一一，1！！！.户口，刘启，口，0，？？？，二进制？
看密文，发现户口是重点，空格是隔断，我先把有户口的字符串替换为0，没户口的替换为1，拿去解密，无果？哪里出了错？
朋友做的相反，是吧有户口的作为 1 ，没户口的作为0，解密拿到了flag

```
01100110011011000110000101100111011110110100100101010011010000110100001101011111010101110100010101001100010
```

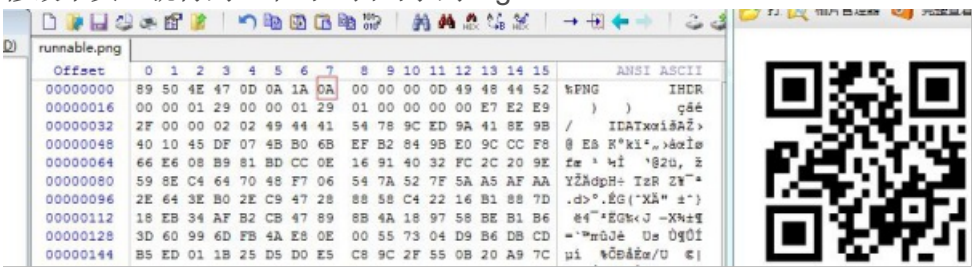
长条多出来了，删掉，二进制转字符串：得到结果

Misc他们能在一起吗

下载附近是个二维码后缀png，我觉得没那么简单，习惯性winhex16打开看，果然里面有压缩文件，打开，发现需要密码。
扫下二维码，信息是：UEFTUyU3QjBLX0lfTDBWM19ZMHUlMjElN0Q=，
Base64明文PASS{0K_I_L0V3_Y0u!}，群里好多人拿到这个就当flag提交还在群里问什么格式，怎么提交不对。
Png改成zip后缀，打开，解压文件，密码输入“0K_I_L0V3_Y0u!”，拿到flag

Misc无法运行的exe

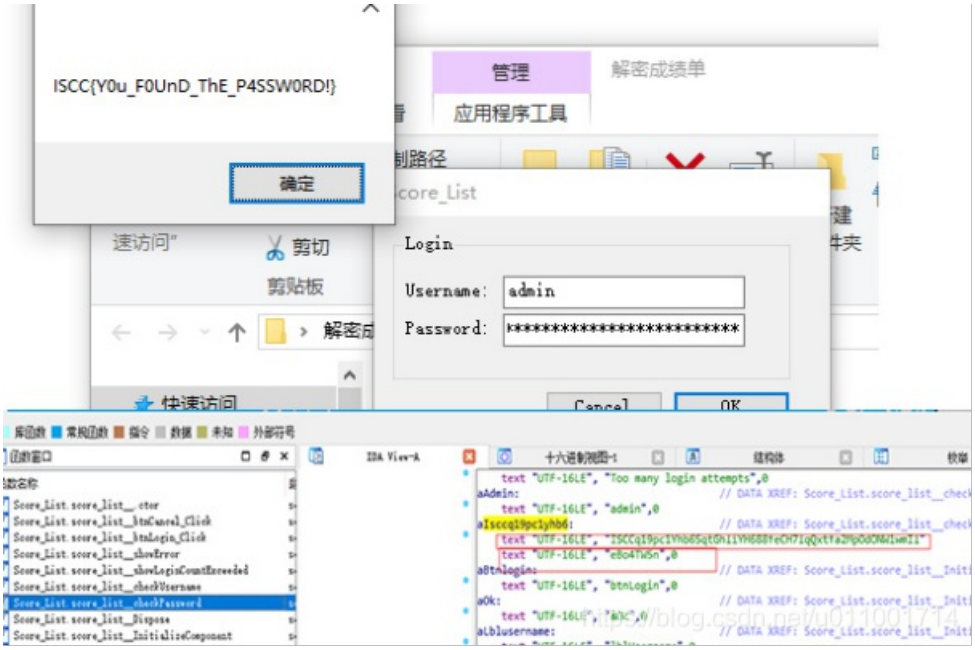
附件里是个exe，winhex查看，明明是一串base64密文，拿去解base64，是乱码。
开头有png字样，想到了是不是图片转base64的。Exe文件winhex打开，编辑》转换文件》 base64->binary,保存成png，打开发现无图像，查看文件头
文件头是89 50 4E 47 0D 0A 1A 00
PNG文件头是89 50 4E 47 0D 0A 1A 0A
修改下头，就得到二维码，扫码拿到flag



Misc解密成绩单

附件打开是个exe，文件头显示确实是exe，运行，需要帐号密码，拖进IDA分析
找到了admin像是用户名，下面字符串，有两行，我复制了第一行，怎么试都不对，后来两行都复制了连在一起作为pass，成功获得flag

```
Text ISCCq19pc1Yhb6SqtGhliYH688feCH7lqQxtfa2MpOdONW1wmIl
Text eBo4TW5n
Pass：ISCCq19pc1Yhb6SqtGhliYH688feCH7lqQxtfa2MpOdONW1wmIleBo4TW5n
```



Misc High起来

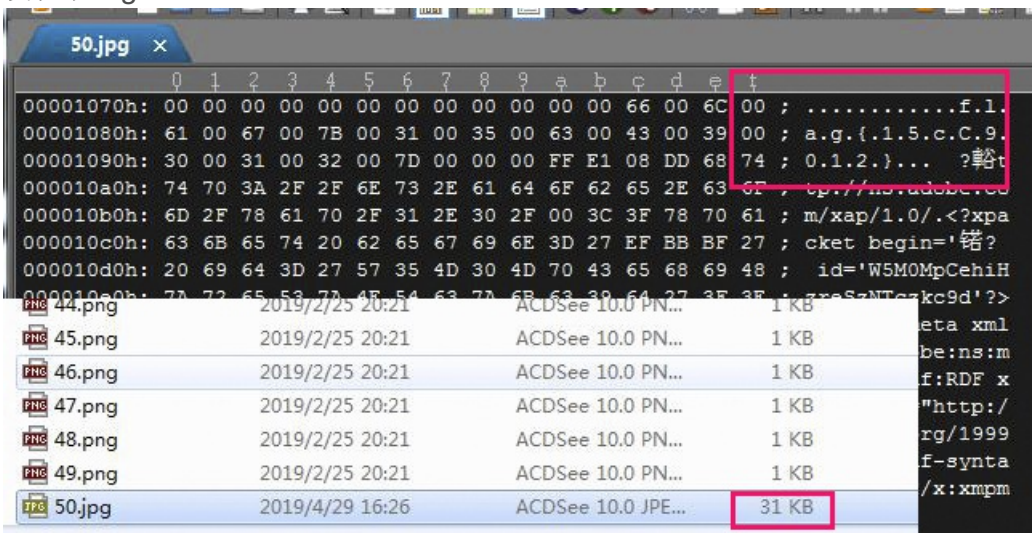
附件里是个png，没图像，发现末尾有1.MP3，一定又是包含了压缩文件，png改成zip后缀，打开发现01.MP3
看到MP3，专门去百度了音频隐写术，音频可视化工具也打开看了没有信息。
MP3Stego打开，MP3Stego -x -p pass 1.mp3，不知道没密码，用不了。后来经经提醒让我再看看png图片为啥不显示。
果然又是文件头损坏，改文件头89 显示二维码图片，扫码拿到当铺密码
Pass：中口由羊口中中大中中中井
当铺密码解码：`201902252228`
然后 `MP3Stego -x -p 201902252228 1.mp3`
解出信息放在txt中，拿到

```
&#102;&#108;&#97;&#103;&#123;&#80;&#114;&#69;&#116;&#84;&#121;&#95;&#49;&#83;&#99;&#67;&#57;&#48;&#49;&#50;
```

然后Unicode转ascii就是flag了

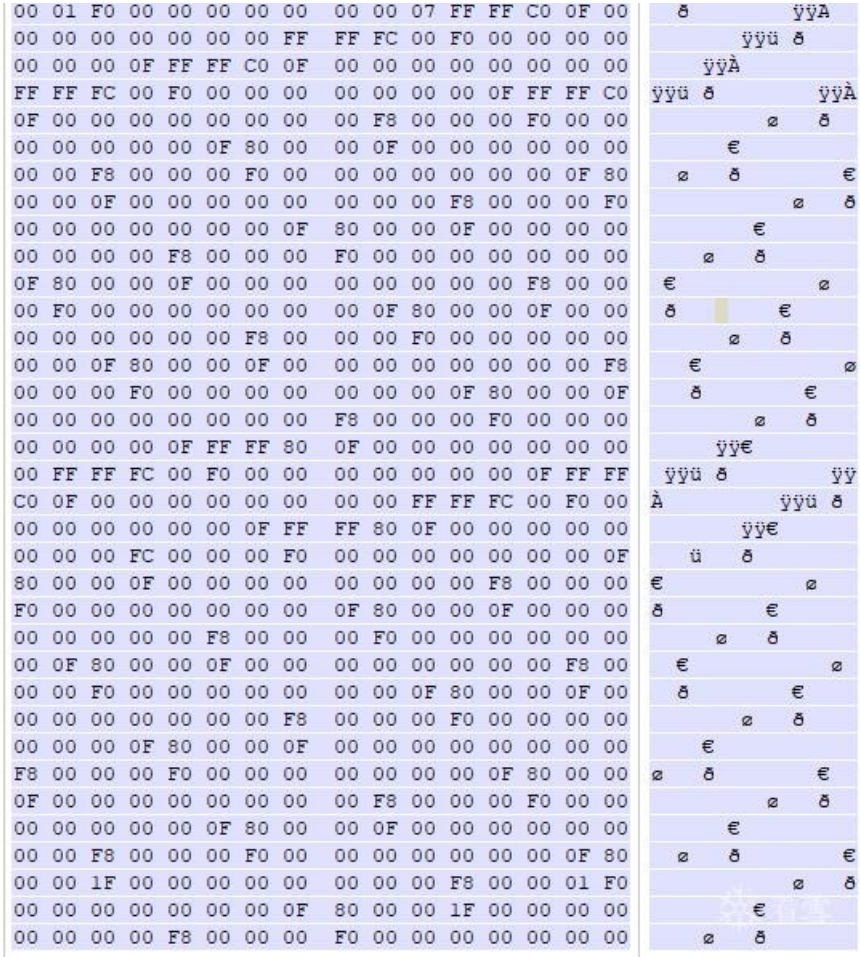
Misc最危险的地方

下载附件：Misc-01.zip
发现里面一张jpg图片，打开无图像，16进制看到存在压缩文件头，以及50个图片名
改后缀为压缩文件或者直接winrar打开，解压，全是二维码，扫码。
由于想抢百血，一直很紧张，思路也不清晰，一直在和二维码较劲，都没仔细看文件大小和后缀名.
后来百血没了才冷静下来发现第50个图片35kb，其他的图片也才8百多字节，仔细一看后缀是jpg，拖进编辑器发现了flag。



Misc 碎纸机

找到了八张图片时，懵了，有大佬暗示要用opencv，或者gl，可是都不会用后来遇到大佬给思路，说是每个图片后边插入的数据，很有规律，



选块复制16进制，16进制数粘贴到文本，然后会发现字母轨迹，照做，费力凑出flag。

Mobile

Mobile01

安卓逆向，安卓不了解，反编译了下看了java无果。
自己云里雾里。朋友提示我用check1爆破apk的 so文件。
然后拿到：`1234567836275184`

Reverse

RE-answer to everything

IDA打开，就发现这个可疑，当作flag提交无果，想到题目上的提示sha1，
我就把这段字符加密，#kdudpeh进行sha1加密，做flag提交不对
去掉了#，加密80ee2a3fe31da904c596d993f7f1de4827c1450a
套上flag｛｝，正确得分。

RE-dig dig dig

原题，TWCTF的dec dec dec。
思路主要是base64 加密 rot13加密 uudecode加密
我们拿flag就需要反着来一边。
文件在IDA打开，shift+f12打开字符串窗口，找到一串密文：
@1DE!440S9W9,2T%Y07=%<W!Z.3!:1T%S2S-),7-$/3T
然后uudecode》rot13》base64 就可拿到flag

```
密文：@1DE!440S9W9,2T%Y07=%<W!Z.3!:1T%S2S-),7-$/3T
Uudecode解
密文：FIAQD3gvLKAyAwEspz90ZGAsK3I1sD==
ROT13解
密文：SVNDQ3tiYXNlNjRfcm90MTNfX3V1fQ==
base64解
结果：ISCC{base64_rot13__uu}
```

当然，还有python脚本：

```
targetString = "@1DE!440S9W9,2T%Y07=%<W!Z.3!:1T%S2S-),7-$/3T "
decodedBits = ""
for c in targetString[1:]:
    decodedBits += bin(ord(c) - ord(' ') + 64)[-6:]
decodedText = ""
for i in range(0, len(decodedBits), 8):
    decodedText += chr(int(decodedBits[i:i+8], 2))
print(decodedText)

rot13edText = ""
for c in decodedText:
    if ord(c) >= ord('a') and ord(c) <= ord('z'):
        rot13edText += chr((ord(c) - ord('a') + 13) % 26 + ord('a'))
    elif ord(c) >= ord('A') and ord(c) <= ord('Z'):
        rot13edText += chr((ord(c) - ord('A') + 13) % 26 + ord('A'))
    else:
        rot13edText += c
print(rot13edText)

from base64 import b64decode
print(b64decode(rot13edText))
```

Rev04它被flag污染了

IDA也不好使了，发懵，然后看到其他re类的题。在linux输出下可视的字符。
咱也死马当活马医。开虚拟机，kail打开
命令：strings bad
输出了好大一堆，我就慢慢一行一行看，终于一串有大小写和数字的字符串引起我注意

```
dWdnYzovL1ZGUFAyMDE5e2hleV9mcnJ6Zl9lYmdncmFfanZndX1wZ3MucGJ6:
```

复制了，感觉像base64，解码一下。

```
base64 >
uggc://VFPP2019{hey_frrzf_ebggra_jvgu}pgs.pbz
```

什么东西？某种协议？
内容会是｛｝里的？
想到ctf惯用加密，Rot13试了下

```
rot13>
http://ISCC2019{url_seems_rotten_with}ctf.com
```

整体提交

----------未完待更----------