ZJPC-CTF_2015.11~12_writeup_by_GoldsNow

原创

GlodsNow 于 2015-12-17 21:39:33 发布 1298 收藏

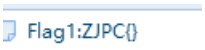
分类专栏： writeup 文章标签： ctf zjpc

本文链接：https://blog.csdn.net/GlodsNow/article/details/50347041

web：

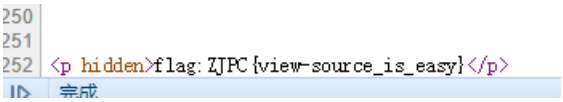
（1） 签到题：

打开网页看标题就可以了。。送分的。



（2）你会看源代码吗：

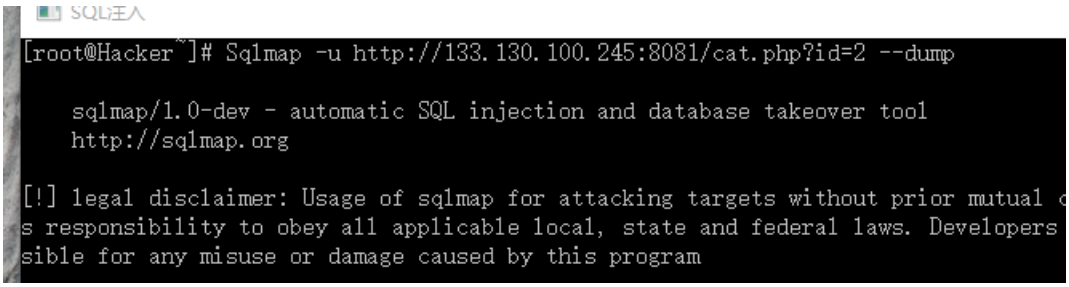
右键查看源代码！在最下面就



就能找到flag

（3）管理员的密码是什么：

sql注入 –u url –dump 一下



```
[14:30:34] [INFO] analyzing table dump for possible password hashes
recognized possible password hashes in column 'password'. Do you want to crack them
q] n
Database: photoblog
Table: users
[1 entry]
+----+-------+----------------------------------+
| id | login | password                         |
+----+-------+----------------------------------+
| 1  | admin | 9ca1a6ff3663161161f7493a6872f301 |
+----+-------+----------------------------------+

[14:30:37] [INFO] table 'photoblog.users' dumped to CSV file 'F:\CTF\??\SqlMap?Pytho
```

一眼看出password是MD5码 找个MD5解码网站 直接能解密出来。

（4）OMG:

在（3）基础上登录那个网站

Title:
File: 选择文件 未选择任何文件
test
Add

在这个地方上传含有一句话木马的php。后缀改成Php之类的就可以了（利用后缀名的欺骗o(∩_∩)o 哈哈这是问袁大神的），然后再在菜刀里打开后门 就能进入 然后再根目录里找到flag.txt
flag就在那里。

（5）AAENCODE:

打开aaencode的网页查看源代码了解其编码规律
或者笨办法一个一个字母凑，这样也能出结果。
我看那个源代码看奔溃了 然后就用笨的办法做的。然后也感觉这个每一个图案其实有所对应的数
有一点 Ⅳ Ⅴ Ⅵ 这种数字的感觉，。

（6）Easy_BYPASS

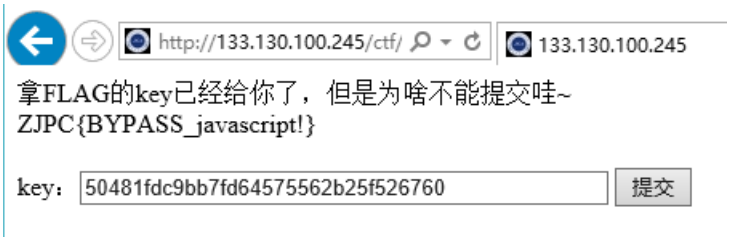
先查看源代码，要绕过本地验证，把源文件复制下来用记事本打开来修改，具体修改方式可以查阅百度←点击这里，修改action为整个的网址、去掉onsubmit=”return check(); 然后打开html文件就可以了。

```
        <div>
"form" action="1.php" method="GET" autocomplete="off" onsubmit="return check();">
<input name='pwd' type="text" size="50" value="50481fdc9bb7fd64575562b25f526760"/>
t type="submit" value="提交">
```

只需要修改这两个就可以了；

```
f细看看表单提交的代码吧，你会有意想不到的收获 -->
v>
      <br/>
      <form id="form" action="http://133.130.100.245/ctf/1.php" method="GET" autocomplete="off" >
              key：<input name='pwd' type="text" size="50" value="50481fdc9bb7fd64575562b25f526760"/>
              <input type="submit" value="提交">
      </form>
iv>
```

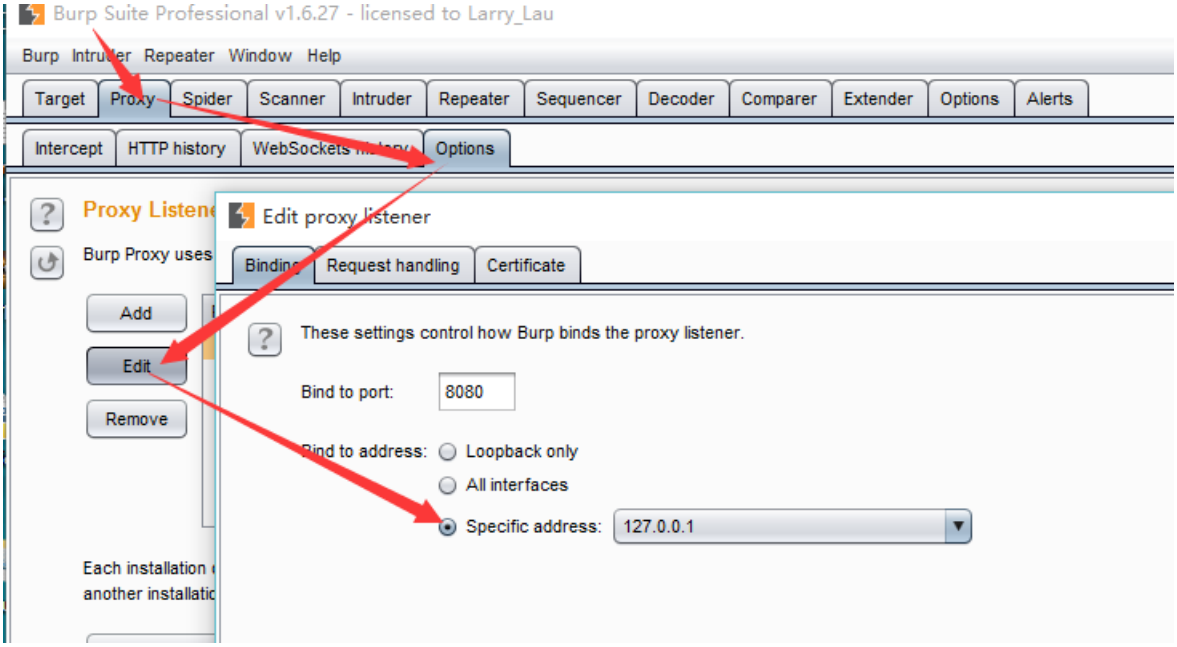
然后保存为htm文件 在浏览器里打开，点击提交 答案就出来咯；



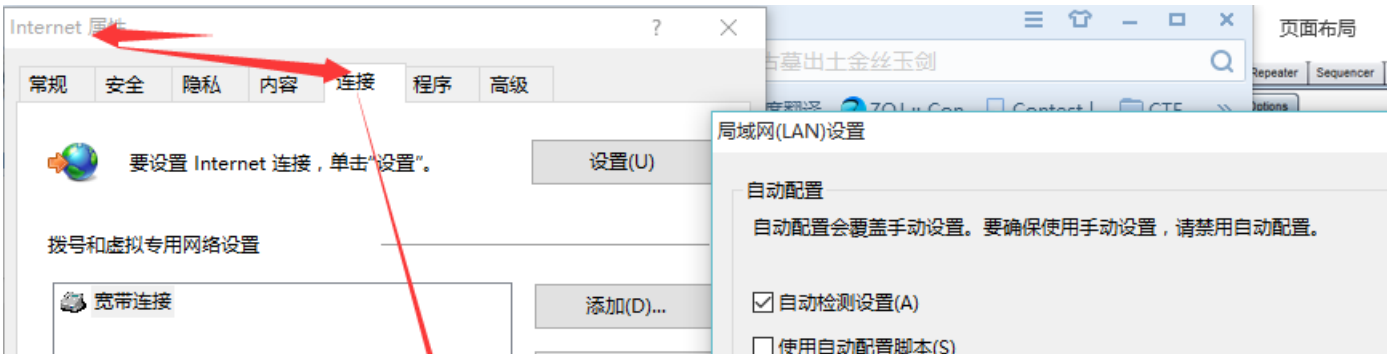
（11）Crack：

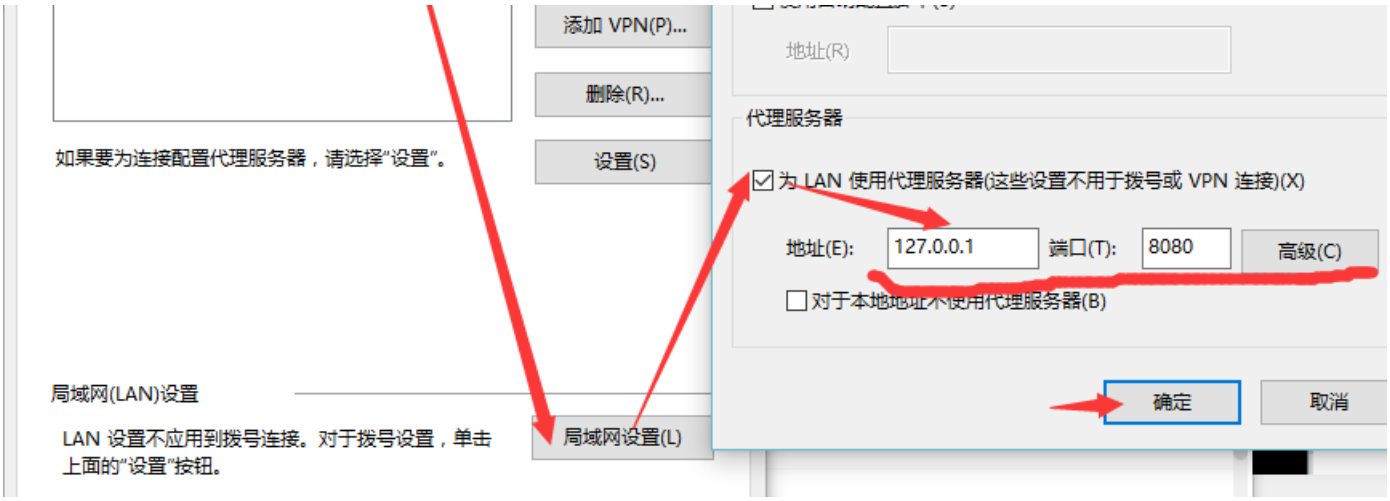
这题是我觉得最有意思的一道题目了。
看了源文件之后知道帐号是xiaoming
然后要密码从1000凑到9999我知道一个一个去输是不现实的，然后就用了Burp

先设置代理；

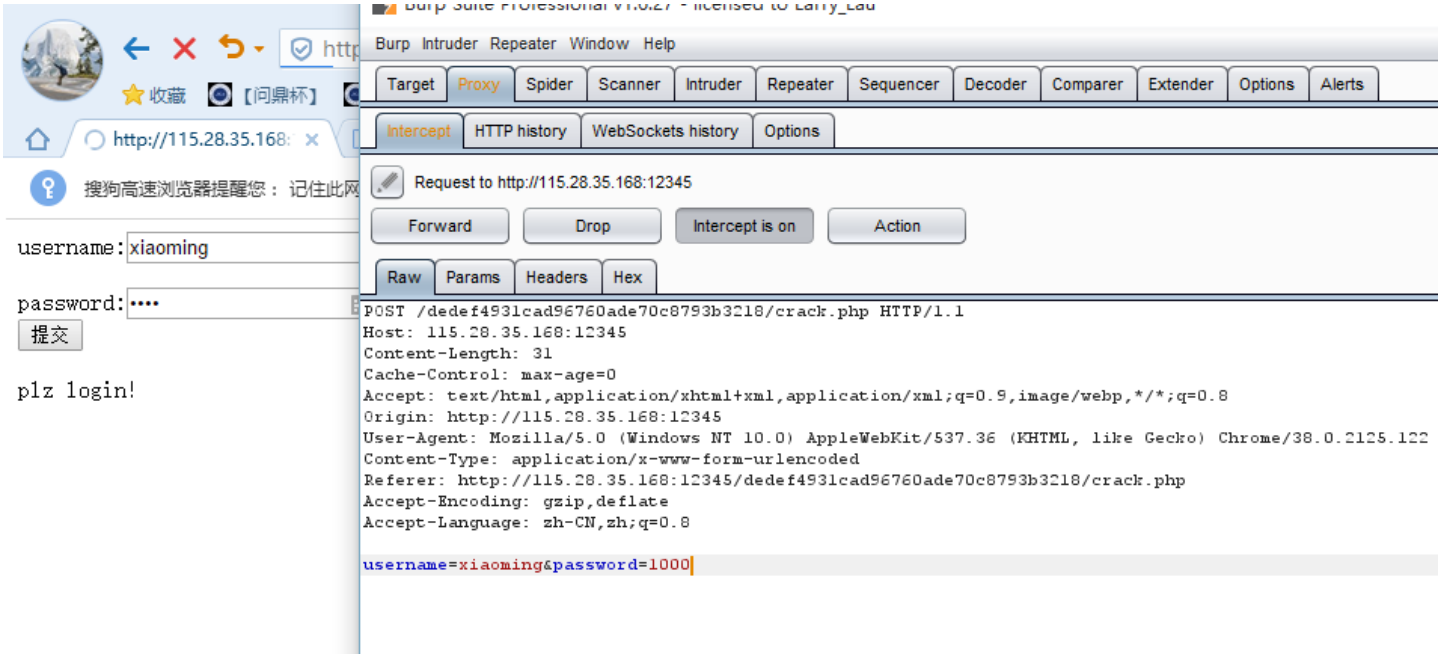


然后打开internet选项设置；

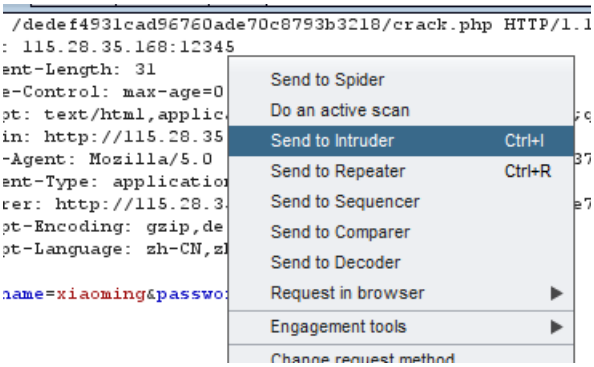




到这个界面后选择intercept on
然后在网页上输入 xiaoming 和 随便的密码 点击提交会发现 burp里已经有些文字了



右击选择 这个。。



Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /dedef4931cad96760ade70c8793b3218/crack.php HTTP/1.1
Host: 115.28.35.168:12345
Content-Length: 31
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://115.28.35.168:12345
User-Agent: Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.122
Safari/537.36 SE 2.X MetaSr 1.0
Content-Type: application/x-www-form-urlencoded
Referer: http://115.28.35.168:12345/dedef4931cad96760ade70c8793b3218/crack.php
Accept-Encoding: gzip,deflate
Accept-Language: zh-CN,zh;q=0.8

username=xiaoming&password=§1000§
```

Add §

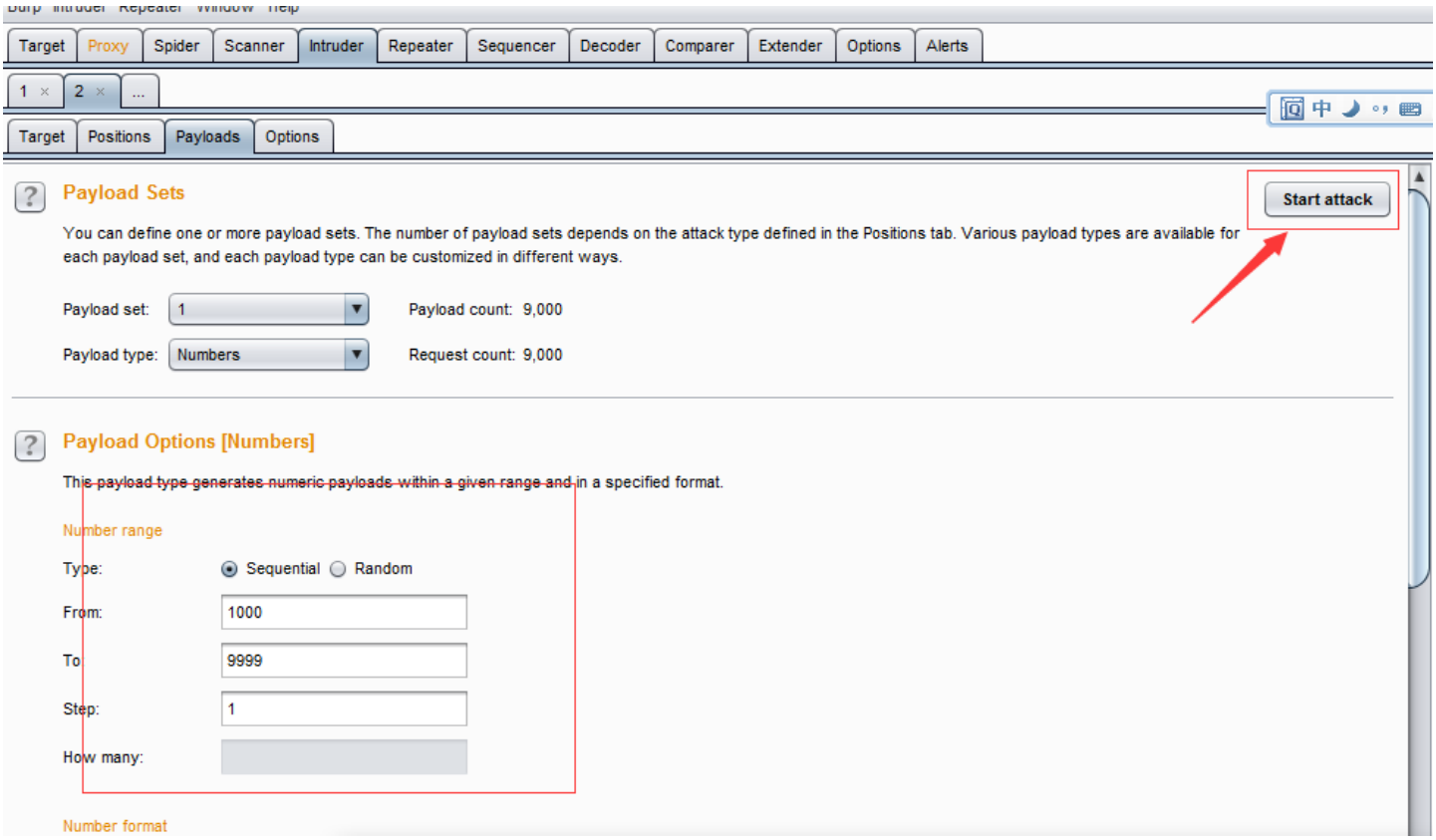
Clear §

Auto §

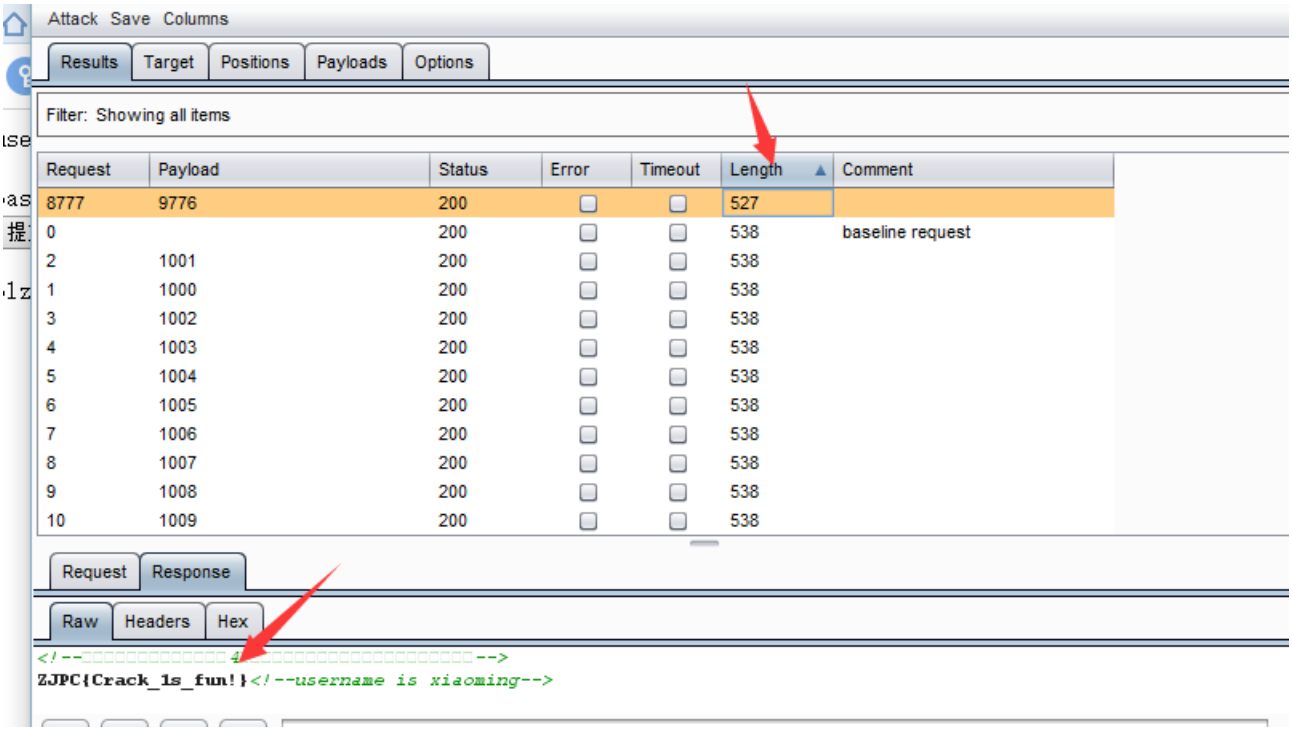
Refresh

先选这一个然后在password 后面的1000选上
点击add

然后设置要变化的数据 然后 点击start attack 开始爆破



这个题目很简单哦，。点下length看下长度不同的那个就是正确的 然后在返回源文件的地方就可以看到flag了



---

stega:

（1） Hex!

按提示 在winhex中将图片打开，然后在右边就能看到flag。

```
the hex-edit rev
ealed: Flag:ZJPC
{the_basic_stega
no}
```

（2）LSB

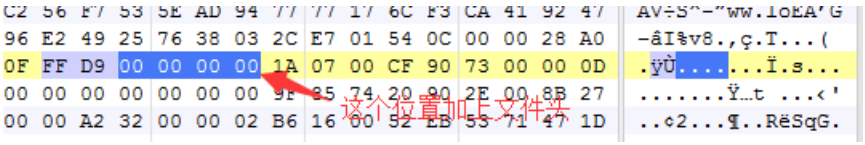
也是单纯利用工具StegSolve 按打开文件后按下面的左右键。

（3）JPG_ OR _RAR?

把后缀.jpg 直接改成 .rar打开就是了，会有一个docx文件里面包含密码。

（4）RAR?RAR!!!

这题。。。我的错。刚开始做法时 把这个改成.rar打开会提示文件损坏，就在winRAR中打算将这个rar文件修复 可是却提示有密码！可是这个又不是逆向题要找到密码然后就在网上下载了一个强力修复软件 一修复就好了，修复后的文件和（3）的内容差不多。
后来的我才发现是我太天真了。这个题目是把rar的文件头去掉了。。
Jpg文件大多是FFD8是文件头 FFD9是文件尾/ 找到FFD9然后再后面加上rar的文件头这个rar文件就好了



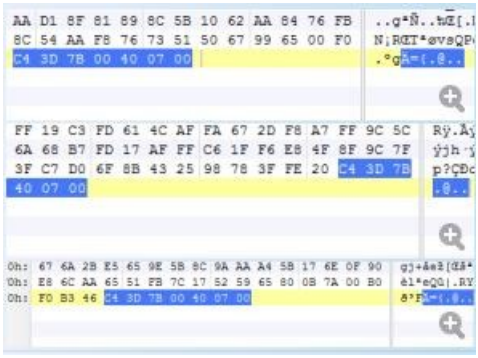
（5） base64?

我是在这里看的，就是这样子的，每排的BASE码都能隐藏2或4个二进制
点击这里 这个是Ulin大神写的write up 可以参考一下
因为不会用python的我写了个C（可以根据 `=` 的数量或者 `\0` 来判断最后几位数）
将这一些二进制的文字连起来

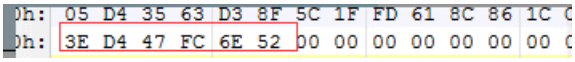
最后转成ascii flag自然而然出现了。。

（6） Steeeega~

解题思路是（3）差不多的，。。。加上Rar！后能打开rar 但是出现密码。。一看那里写着PASSWORD：steeeega 就可以给到提示。。对 压缩包密码就是这个；
打开压缩包后有一张照片。。对，，这个图片就是一个坑。。。没什么用。
右边很显眼的ISA 然后试一下密码是不是 ZJPC{ISA}结果是错的。。
然后我就忽视注释去看别的内容了，，，把重心放在图片上了。。。就是在这里我发现了原来图片的文件头和尾是 FFD8 FFD9 又觉得是隐藏在了RAR文件
对比很多rar的文件 发现一般都是



都是C4 3D 7B 00 40 07 00

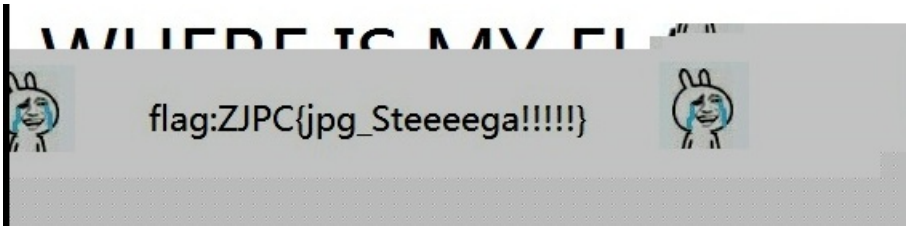


这个rar却与众不同。。。以为是000000后面的那一串16进制能转化出flag来。。然而又跑偏了。。。最后经过hint才知道是在注释。。我脑洞不够大。。。
然后把注释复制到txt 然后就直接变成白色界面因为最后留了一排超长的空白，而且按右键移动的时候还是跳跃式的，，一猜就知道是在这个地方了，
用记事本把短的那个替换成1长的替换成0
然后就是一串2进制的数据 是8位的2进制。。转化一下能出结果。

（8） Jpg_stega

这题挺简单的。。。。只是把第二张的内容复制到第一张下面两张图片合成一张了。
用winhex 或者010打开它
然后看看文件头 文件尾都正确 感觉像是一张正常的图片。。
然后我把从后面开始删数据。然后发现即使删了挺多数据，图片的内容都不会变（会去删数据是因为我做steeeega~被坑过。。研究rar里面的图片研究了好久。然而发现却不是在图片里。。好桑心。。然后就删过下面的数据。。一删就和下面的图片一样的感觉），说明这张图片下面隐藏了很多的内容；
然后删到一定的位置的时候，，

图片开始有地方出不来了，可以说明这张图片差不多就这么多数据；然后记录位置，，然后从这个点开始删数据 向上删，，就能把下面的图片数据显示出来了 删啊删啊删啊



答案出来了 哈哈哈哈哈

Crypto:

（1）Caesar:

这个太简单了 前面的AKQD就提示答案了 与ZJPC都差一位 都减一位答案就出来了 手动都可以。。A→Z K→J Q→P D→C .....

（2）（3）（4）

网上找下转码工具就好了 不想写write up

（5）caesar的复仇：

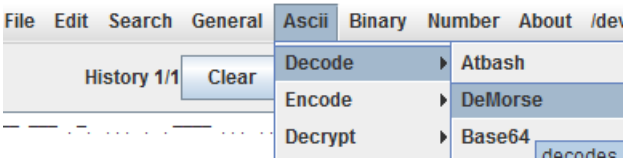
有点意思，就是16进制的数字改成10进制的数字，在用Caesar的方法找出flag。
那时候不懂可以用循环，突然想到密码肯定包括ZJPC 然后就借助着ZJPC之间的
asc差-16 6 -13 找到相差的数字就可以知道ZJPC的位置和这个10进制的
数字与密码ASCII所对应的数字相差值，记得应该是60，把每个数都加60就可以了。
其实这个题目就加一个循环把所有的可能性都一次数出来就可以找到密码的。

（6）编码的集体复仇：

一看数字就是要16进制的ascii解码，然后出来的值后面有“=”这一般就是base64的标志，用base64解码一下，然后就有很多%%%%%，就用urldecode解码一下就好了。

（7）MS:

这是利用morse解密，刚开始的字母明显就是用base64解密的！！后面出现的 -.-. 解密应该不困难。Morse解密在JPK里就能够做到。。

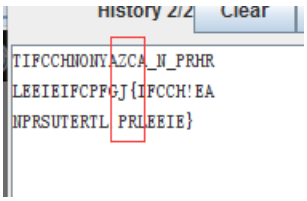


（8）RFC_4648

看了RFC_4648协议就知道用base64解密后base32再解密一次。但是Base32解码有点难找。。

（9）栅栏

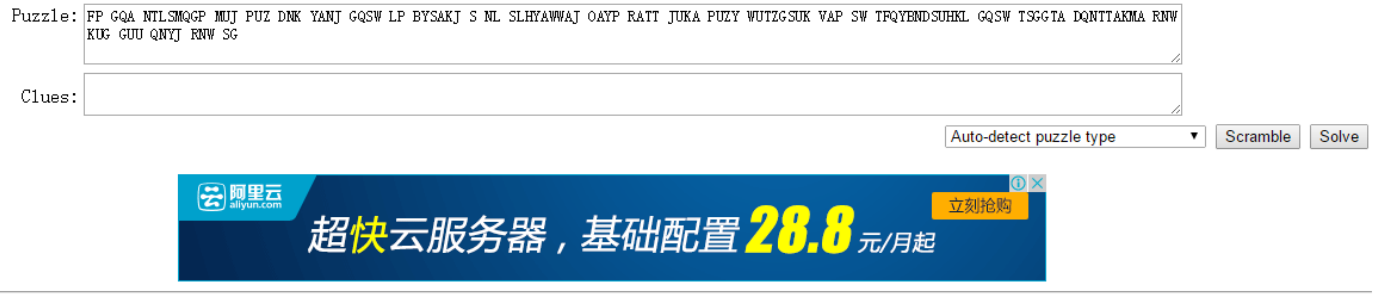
百度一下栅栏的原理，栅栏有2栏3栏的，可是试了一下都不是，那也可能是用加密的方式解密，我把数字等分成3组，用JPK神器能放便挺多。

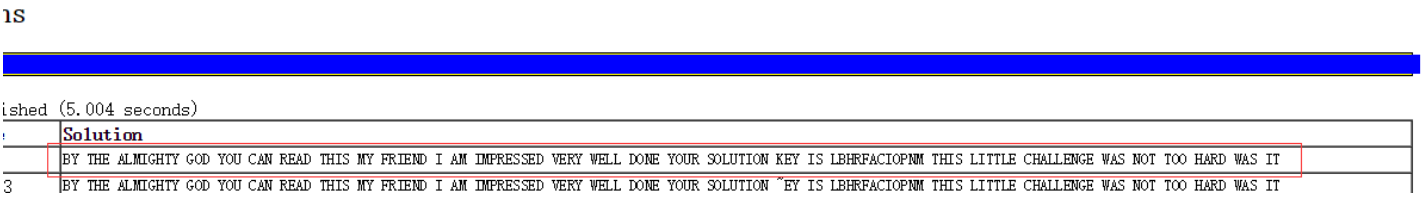


这样顺利发现ZJPC{} 接下来就easy了。

（10）substitution：

用到了一个网站http://quipqiup.com/index.php这个确实挺不错





密码就在下面。。。我知道这个网站看到过这段文字好几次。。就是没仔细看内容。。。果然我对英文没感觉

（11）transportion

两个一组 互相换位。空格也和字符换位置。。写个C或者手动都可以

（12）BIN_coding

一样用JPK分组，一个一个试，发现在7个一组的时候，正好是一组1位的二进制码，用ASCII解密的方法可以解密出来。

（13）keyboard

这题在键盘上看一下位置就好了。

（14）easy_crypto

这题比较坑。。因为要key key就是k[]那个位置的字符串；
因为知道思路，，但是做出来比较难，因为写不出一个程序直接能够把key写出来，，，能力不够，就是以为有k[i % strlen(k)] ^ t，，然后我只能写循环一个一个的凑，，渣渣程序就不亮出来了，，最后被坑了一把。。以为msg001那个最后是有回车键的，，我把它删掉自己添加。。。。然后因为这个密码老是错误，，，一直错，，，而且还找不到原因。。。后来重新下砸了一下文件才把它解决，不开心。然后得到key就是 VeryLongKeyYouWillNeverGuess
再改一下原来给的那一个程序，

```
        printf("Error\n");
        return 0;
    }
    char k[] = "VeryLongKeyYouWillNeverGuess";
    char c, p, t = 0;
    int i = 0;
    while ((p = fgetc(input)) != EOF) {
        c = (p - (k[i % strlen(k)] ^ t) - i*i) ;
        t = c;
        i++;
        fputc(c, output);
    }
```

不难，，三个地方就可以解决的，，。。现在就是想知道如何简单的得到key的方法。。估计也是几部程序能解决的。

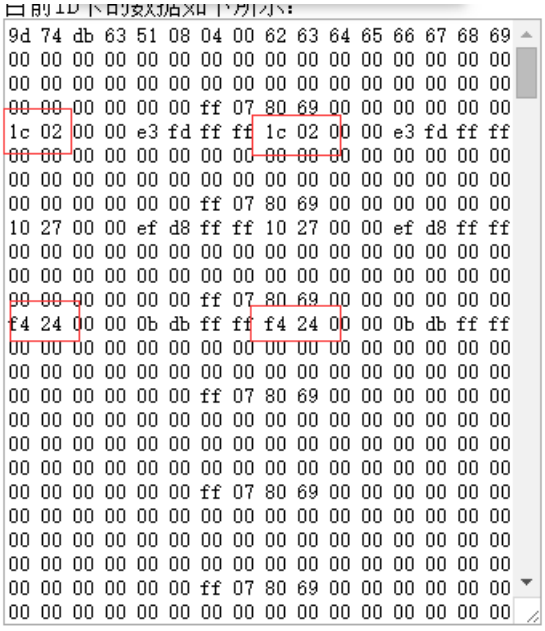
Misc:

（1）我是送分的

这一题比较难，要打开QQ找到 信息安全社【ZJPC-ISA】 这一个群，然后就很神奇的发现。。群号是 329507009！！！

（2） RFID

奔溃。。。做出来后才知道010那么简单。坑爹的我用了winHEX和excel
但是感觉用excel做题好有成就感，。。。。
这是一道有意思的RFID卡解密的题目，主要是要能发现验证的时候有数字是不同的，说明他也是一个刷卡机！但是刷卡的同时只把刷卡后的数值亮出来没有修改卡的数据。



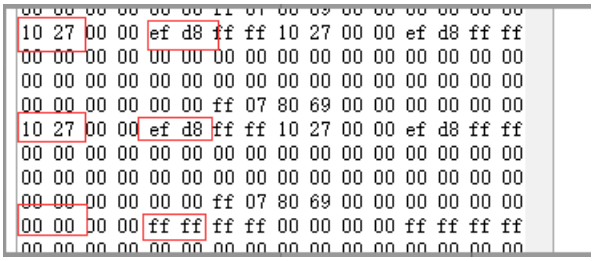
这几个数值是不同的，然后在把这些数值替代到原有数值中（当时我用的是winhex），这样就把钱改了。多重复几次，获取不同金额的时候的值。然后进行研究，我将这几组数值转为10进制的来看，在excel里处理比较方便。

剪贴板 | 字体 | 对齐方式 | 数字

K21 =G21-G24

	F	G	H	I	J	K	L	M	N	O
20										
21	96.4	104	1	151	254	76	-1	-76	1	
22		168	37	87	218	-76	1	76	-1	
23										
24	94.6	28	2	227	253	-180	0	180	0	
25		244	36	11	219	180	0	-180	0	
26										
27	92.8	208	2	47	253	76	-1	-76	1	
28		64	36	191	219	-76	1	76	-1	
29										
30	91	132	3	123	252	76	-1	-76	1	
31		140	35	115	220	-76	1	76	-1	
32										
33	89.2	56	4	199	251					
34		216	34	39	221					
35										
36										

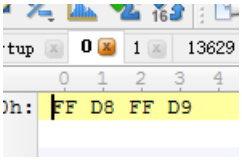
对不同金额的值进行相差比较 发现都是变化差不多的。重点是180那一组，然后推断出其实每组数据实际上+180的处理，超过255（FF）的时候右边那个数字进一位（或者小于零的时候右边一个数字减一），180也代表着1.8元，这样以来总额的计算公式就是X=A+B x 256比如第一组数据168+37*256=9640
然后这样搞来搞去最后发现超过100的时候会有负数的出现，这个又是一个坎，我就研究在1.8元的时候看他扣完钱后会变得怎么样。

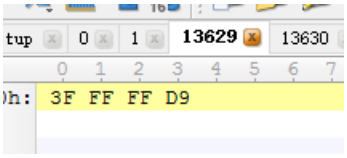


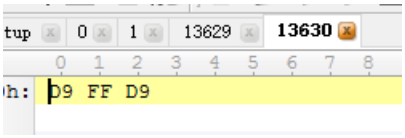
又推断出上面的数据是中间的数据减去下面的数据。也就是说中间的数据10 27就是这张卡能显示的最大的数值，然后把00 00那个位置改成总额为30000
要注意的是上面的每对数字加上下面的每对数字总额为中间的那对数字，，左边一对数字与右边的数字相加总为FF FF
到后来看了hint才知道原来010是多么厉害啊，，只要直接能够显示10000之类的并且可以直接在10000那个地方修改数据了，，，居然没用那个软件 亏大发了。

（4） 把多个文件先放进010

因为JPG的文件头是FFD8尾是FFD9,（图为第0个文件和最后两个文件）一眼看穿每四个字符就俩能用。并且最后一个文件只能用一个字符







所以看到这个文件就是把么多的文件的前两个字符放进同一个文件就好了，
就是每个文件的FFD9是个不用的。多余的干扰项。
不会用python 写了个C，就是把这么多文件的前俩字符放进1.txt

```c
# include<stdio.h>
# include<windows.h>
int main(void)
{
    FILE *fp,*out;  //文件指针
    char ch;
    int n=0;
    char filePath[256];
    out = fopen("2.txt","w");
    for(n=0;n<13631;n++)
    {
        sprintf(filePath,"%d",n);
        fp=fopen(filePath,"rt");   //打开文件
        if(fp==NULL)
            printf("文件打开失败！\n");
        ch=fgetc(fp);      //读取文件内容
        fputc(ch,out);
        ch=fgetc(fp);      //再次读取文件内容 第二字节
        fputc(ch,out);
        fclose(fp);
    }
    fclose(out);
    system("pause");
    return 0;
}
```

然后把1.txt文件的后缀改成jpg
但是有一个问题就是用C语言写的文件会有一个问题就是gets的时候会莫名多出换行符。。这题是问袁大神直接拿的答案 还说是人工审核。。。o(∩_∩)o

---

Program:

（1） Easy_progarm

只需写个简单的编程判断横竖，两斜线，四个四个的数字乘机，就能出密码了。

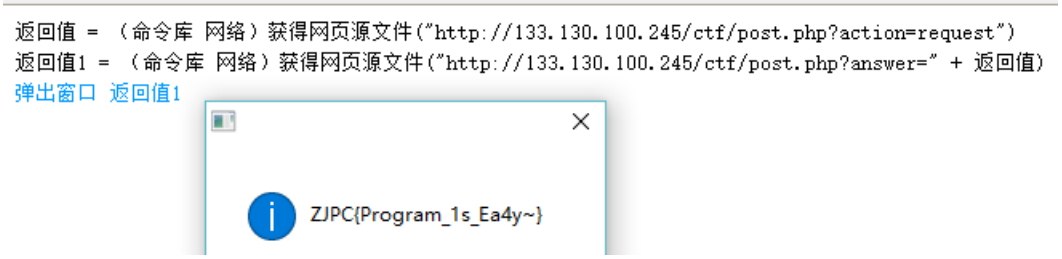
（2） SXBX

太丧心病狂了 刚开始我没做出来。。。看了一下程序肯定会在某个时候循环的。最多100次！！！然后只是找到几个数字一循环。。。好像是59个数字循环一次，然后自己手算出结果。
然后仔细一想就是100000那个数字实在太大了。。运行不过来了把
在VC那里设置一下就好了
工程的设置的连接 再在分类里选择输出 把保留输到最大。。

答案直接出来了

（3） 1.33333333

不会写python 只能用曾经的按键精灵。。然后也是一下就好了 就出结果的。



reverce:

这一块是渣 。。就随便糊弄写一点。。。。指针有点坑

（1） easy_crackme

原谅我当时把exe改成txt。。。找到有｛｝的就当作flag提交然后就对了 。
貌似现在flag变了 惭愧惭愧。。然后没有重新做过。。。

（2）你会PE逆向吗

打开神器IDA 按F5 会出来程序编码。。只要看懂就好了。就是那个指针有点坑，指向的是一段数字里的字母然后就能得到密码。

（3）PEPEPE!

同（2）但就是那个读取语句是读取string的。

（4）ELF!ELF!

用IDA64打开文件，

```
}
v8 = 854698492143LL;
v9 = 880468295913LL;
v10 = 597000454391LL;
v11 = 605590388953LL;
v12 = 932007903423LL;
v13 = 760209211613LL;
v14 = 579820585151LL;
v15 = 940597838039LL;
v16 = 191;
```

重点就是在这里，
而_ DWORD是代表偏移四字节的意思，，然后V8 那边的LL代表的意思是LongLong 长长整型， v17[j] != (((_DWORD)&v8 + j) - 1) / 2
这句话很难读懂，，，然后偏移四位字节“，也不知道然后就想到存储数字都是用二进制的
然后就把这一些数字都转化为二进制，就很容易感受到答案是什么了。。

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
11000111000000000000000000000000011101111
11001101000000000000000000000000011101001
10001011000000000000000000000000011110111
10001101000000000000000000000000011011001
11011001000000000000000000000000010111111
10110001000000000000000000000000011011101
10000111000000000000000000000000010111111
11011011000000000000000000000000011010111
```

```
for ( j = 0; j <= 16; ++j )
{
  if ( v17[j] != (*((_DWORD *)&v8 + j) - 1) / 2 )
  {
    v24 = 1;
    argv = (const char **)((*((_DWORD *)&v8 + j) - 1) / 2);
    printf("%d", argv, v8, v9, v10, v11, v12, v13, v14, v15, *(_QWORD *)&v16);
  }
}
if ( v18 != 48 || v19 != 56 || v20 != 50 || v21 != 51 || v22 != 125 )
  v24 = 1;
```

没错就就是8位二进制这样子来的。
再加上下面的一些数据