AWFUL

萌新劝退周。。。。。

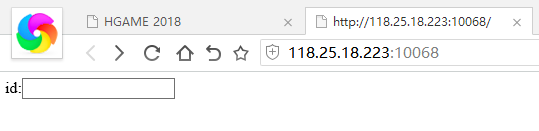
Web

分数100：送分的SQLI

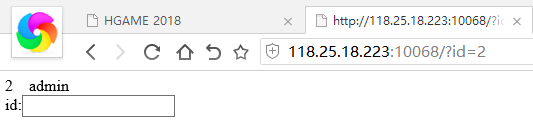
描述：送分题，不解释了

URL：[http://118.25.18.223:10068/](http://118.25.18.223:10068/" \t "_blank)

登陆后的界面



随便输入数字后



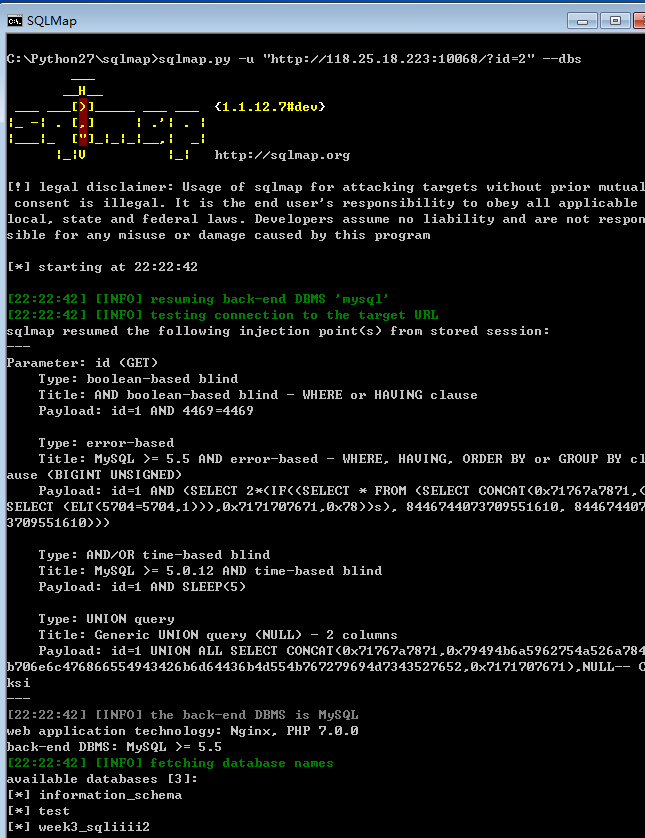
既然是SQL注入，而且可以提交URL中也带有参数，那么直接放到SQLMAP进行扫描注入。打开sqlmap在命令栏输入：

sqlmap.py –u “<http://118.25.18.223:10068/?id=2>”



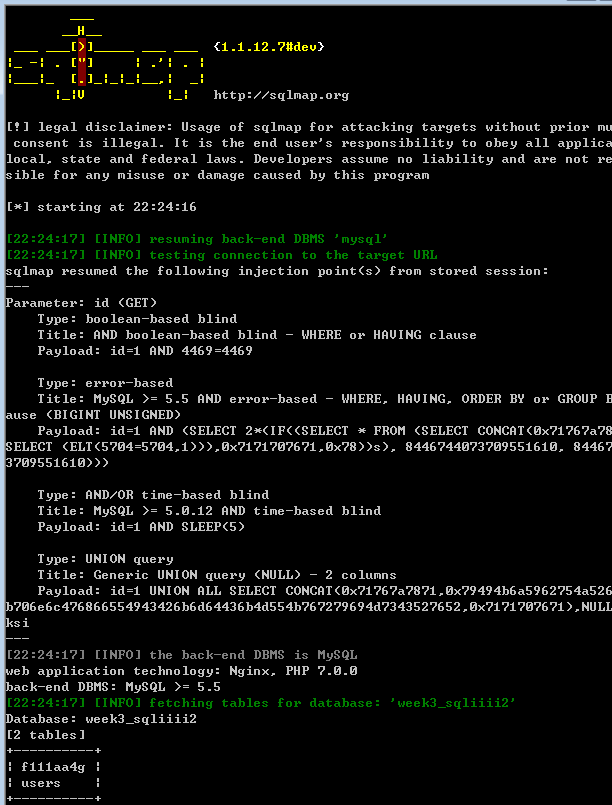
可以看到结果这的确是个注入点，那么继续。

在命令栏输入：sqlmap.py –u “<http://118.25.18.223:10068/?id=2>” –-dbs 来获取数据库名字



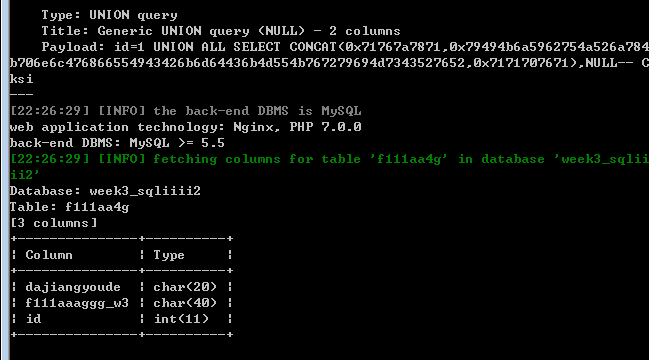
从左下角可以看到结果，猜测flag可能在week3\_sqliiii2中

输入命令sqlmap.py –u “<http://118.25.18.223:10068/?id=2>” –D week3\_sqliiii2 –-table



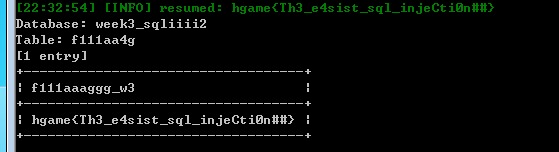
结果出现在左下角，猜测flag可能在f111aa4g的表中

继续输入：sqlmap.py –u “<http://118.25.18.223:10068/?id=2>” –D week3\_sqliiii2 –T f111aa4g –-columns



从结果可以看到flag应该在f111aaaggg\_w3列中.

继续输入: sqlmap.py –u “<http://118.25.18.223:10068/?id=2>” –D week3\_sqliiii2 –T f111aa4g -C f111aaaggg\_w3 –-dump获得flag



tip：刚接触这个软件的人注意下每次输入命令的最后一个选项是两个“-”就行了

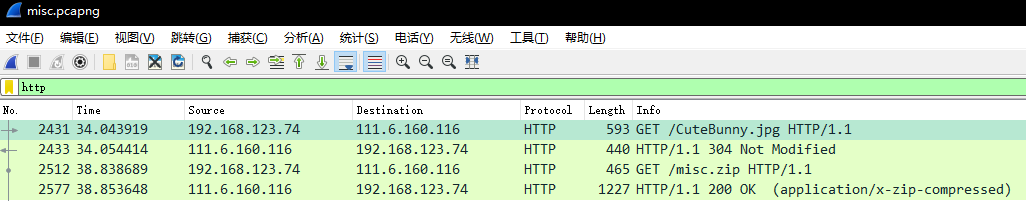
MISC

分数200：bunny treasure

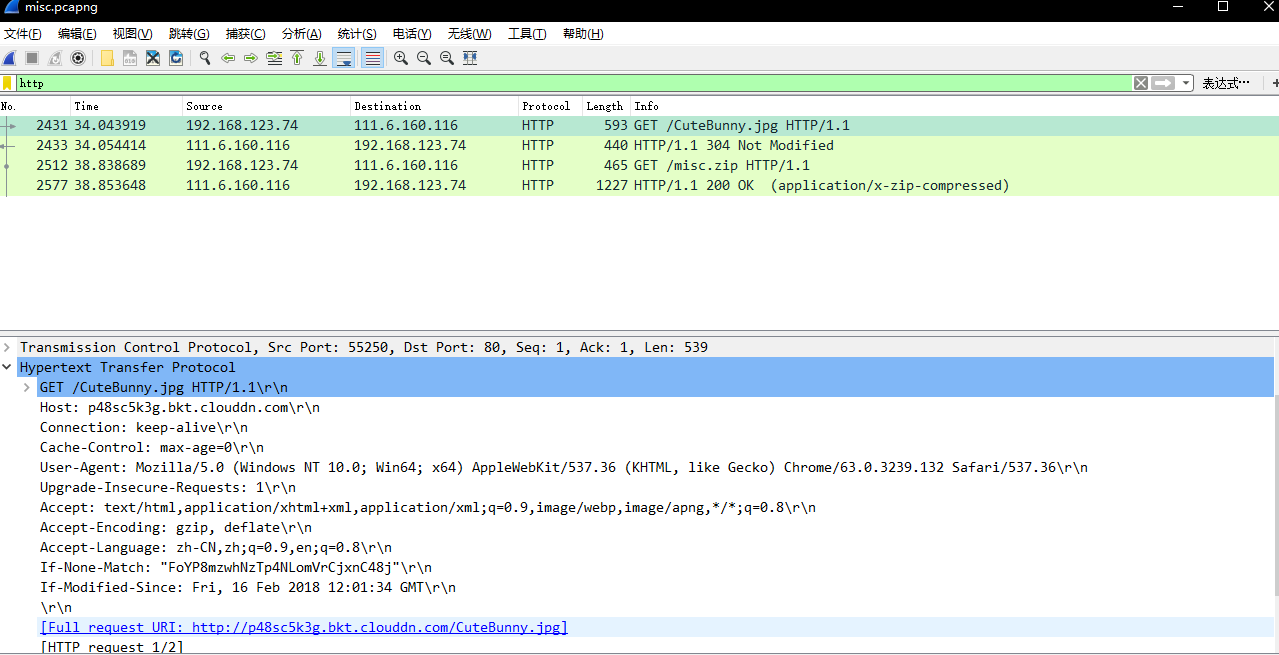
描述：I find a bunny pic.  
And I treasure up it.  
hint: All the clue you want is in it.

URL：[http://p48sc5k3g.bkt.clouddn.com/misc.pcapng](http://p48sc5k3g.bkt.clouddn.com/misc.pcapng" \t "_blank)

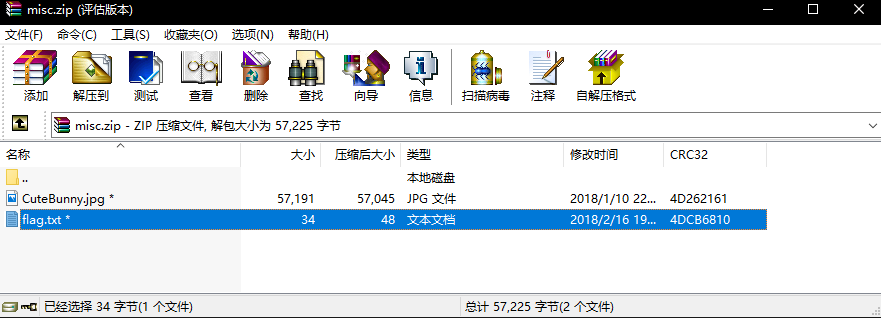
下载过来为一个未加密的流量包，先进性筛选，在过滤器里面输入http



可以看到有两个文件（当初只注意到第一个文件，结果弄了一段时间没有任何结果，后来检查数据包才发现有两个），如图这个位置（以第一个文件为例）可以分别下载到两个文件（一个图片和加密过的压缩包）



打开压缩包可以看到带有flag名字的txt，推测需要解密才能得到flag



由于是zip压缩包，花了一晚上用Ziperello跑密码无果便放弃。查询网上得到思路[1]，第一个想法是伪加密，放到kali linux检查不是，用ZipCenOp.jar再三确认后放弃这个思路。第二个想法是明文攻击，因为从流量包中得到的图片和加密的压缩包里面的图片完全一样，符号明文攻击的条件。下载AZPR软件，将之前的图片压缩到一个单独的压缩包中



可以看到他们的CRC32是完全一样，可以进行明文攻击，用AZPR进行解密



将结果保存下来，打开即可获得flag

hgame{^P1ay\_H9am3\_2nd\_p1Ay\_buNNy^}

题外话：当初写这题的时候，以十六进制查看图片可以在开头看到photoshop 3.0 8BIM的字样，以为是那种flag藏下图层下面，用photoshop打开报错说解析错误，以为是版本问题又下了个新的结果还是不行。用画图打开，另存为一个新的jpg图片photoshop可以打开但是没有进展，莫名其妙的花了很长时间。

参考文章：

[1] <https://www.cnblogs.com/WangAoBo/p/6944477.html>

分数250：画风不一样的她

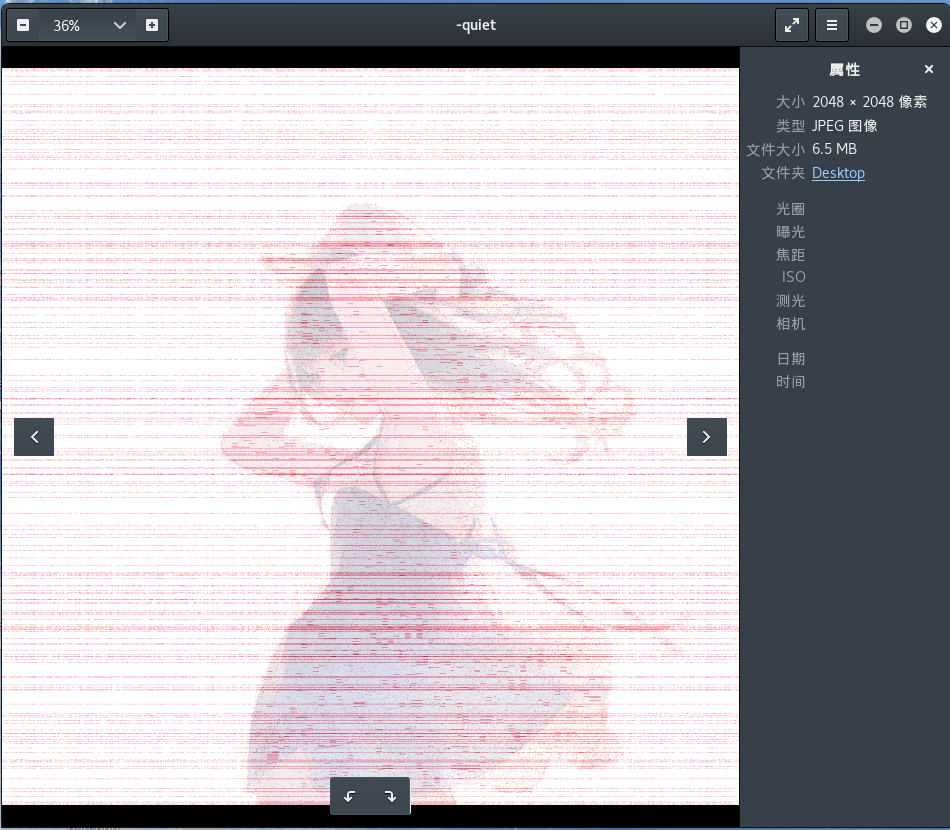
描述：来找不同呀

hint: 盲水印

URL：

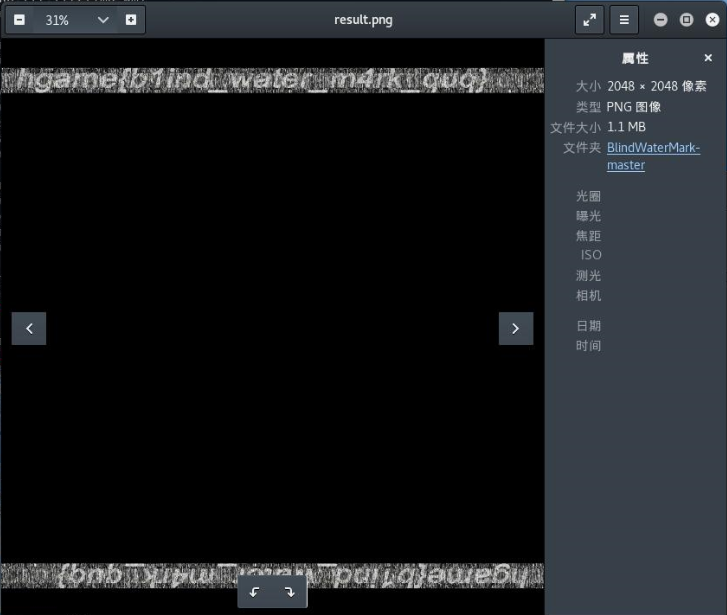
<http://p3pqfvzzm.bkt.clouddn.com/%E7%94%BB%E9%A3%8E%E4%B8%8D%E4%B8%80%E6%A0%B7%E7%9A%84%E5%A5%B9.zip>

下载压缩包得到两张图片0.png和1.png,放到kali linux中检查发现0.png实际上为jpg文件，将后缀名进行修改，用binwalk和Stegsolve对每一张图进行检查无果，LSB隐写也没有进展，之前隐写做过某一道也是有两张图的题目，第一个思路为两张图需要进行sub或者XOR得到新的一张图，但是得到的图片用十六进制打开分析没有什么进展。查询网上，得到一个思路：盲水印攻击，将两个图片放到kali linux中，在命令端输入compare 0.jpg 1.png –quiet（若不带quiet选项则会报错不知道什么原因）



可以看到结果中有很多红线，符合盲水印的特征，下载BlindWaterMark[2],在命令端，输入

python bwm.py decode 0.jpg 1.png result.png 其中result为新生成的文件，打开后的到flag



题外话：当初还没给hint前，一直很纳闷这道题为什么只有8个人写出来，结果现在一放hint做出来的人直线上升。

参考文章：

[1] <https://www.jianshu.com/p/02fdd5edd9fc>

[2] <https://github.com/chishaxie/BlindWaterMark>

分数200：这是啥

描述：我也不知道这是啥 拿到flag后找XXXXXXX换取真正的flag

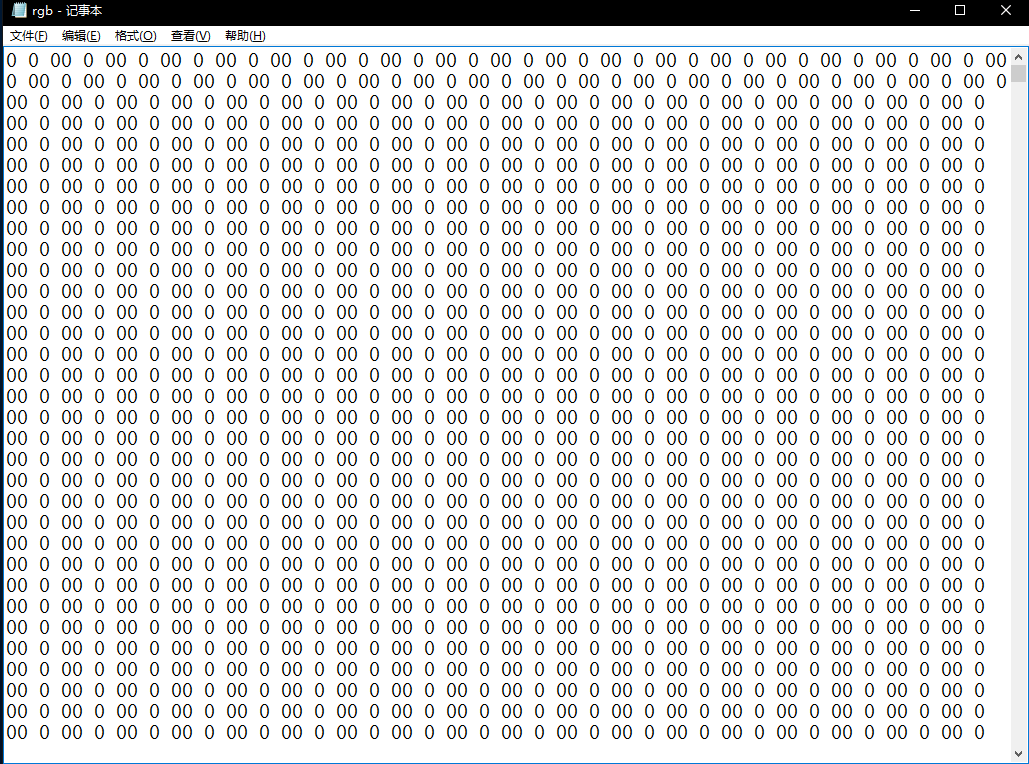
URL：<http://p1kaloi2x.bkt.clouddn.com/rgb.zip>

题外话：因为自己零代码量，所以脚本方面可以自己写。

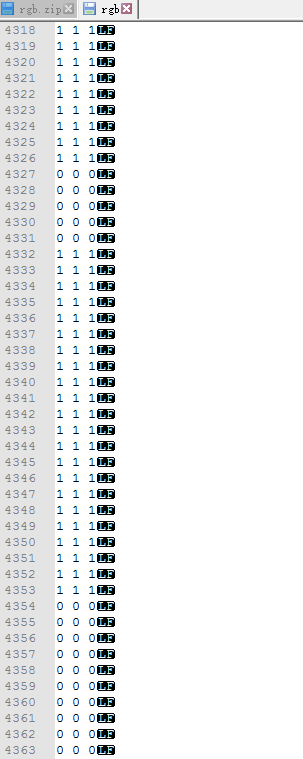
下载得到一个加密的压缩包，确认不是伪加密后直接拿到Ziperello里面跑，跑了一晚上没结果便放弃，用binwalk查看无果，用notepad以十六进制打开压缩包在结尾意外的发现了一个显眼的东西



看这个内容猜测为base64，进行解码得到明文：key is here no one knows:hammernb，其中hammernb就是压缩包密码（当初以为这是flag。。。），打开里面的rgb文件

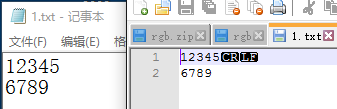


全是0，但是往下翻可以看1，这样貌似太乱了，我们用notepad打开



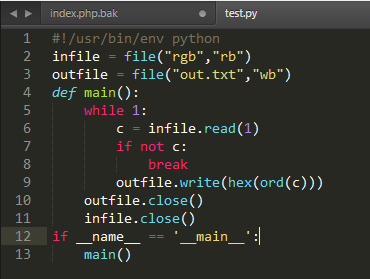
这样好多了。既然标题是rgb，而且最后的行数正好为280的平方，那么这里推测这可能是二维码，而0 0 0 和1 1 1分别代表黑白，接下来需要用脚本进行解析。

这里解释下我们在文本和notepad查看的结果是不同排版的原因。



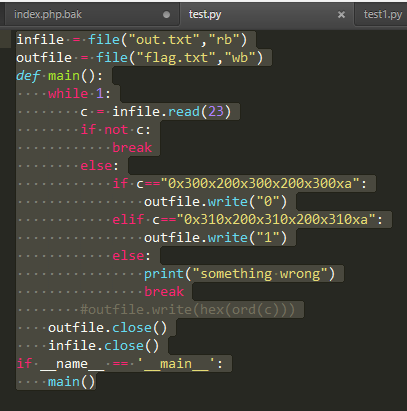
如图，可以看到正常换行他是有两个符号，CR是回车符（相当于\r），LF是换行符（等于\n），而txt在显示结果时需要CR和LF才会执行换行，但是notepad只需要一个（可能会有错误）

先将rgb文件内容转为十六进制，查询得到进行修改得到如下脚本[1]：

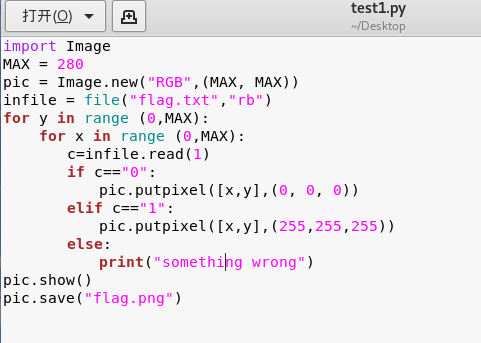


得到新文件out.txt，（由于文件非常大所以不打开了，输出的格式为0x300x300x20….类似的）

第二个脚本（将0 0 0转为0而1 1 1转为1）



第三个脚本则是生成二维码



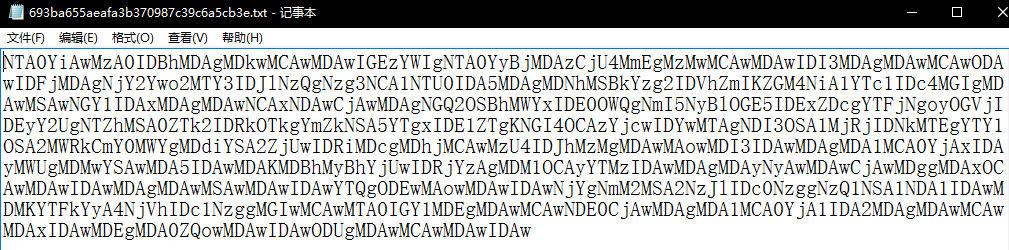
生成新文件flag.png



结果的确为一个二维码，用手机扫描没有出结果，使用QR\_research进行扫描得到结果



通过网址得到一个没有后缀名的文件，拉到kali linux 使用file命令检查结果为txt文本，打开得到一堆乱码



推测为base64，进行解密得到明文

504b 0304 0a00 0900 0000 a3ab 504c c003

582a 3300 0000 2700 0000 0800 1c00 666c

6167 2e74 7874 5554 0900 03a1 dc86 5afb

dc86 5a75 780b 0001 04f5 0100 0004 1400

0000 4d69 a1f1 149d 6b97 e8a9 11d7 a1c6

28ec 12ce 56a1 4e96 4d99 bfd5 9a81 15e8

4b88 3b70 6010 4279 524c 3d11 a659 61dd

f41f 07ba 6f50 4b07 08c0 0358 2a33 0000

0027 0000 0050 4b01 021e 030a 0009 0000

00a3 ab50 4cc0 0358 2a33 0000 0027 0000

0008 0018 0000 0000 0001 0000 00a4 8100

0000 0066 6c61 672e 7478 7455 5405 0003

a1dc 865a 7578 0b00 0104 f501 0000 0414

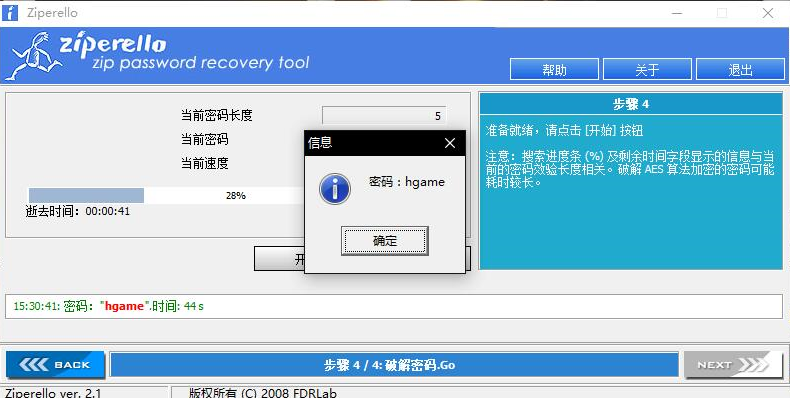
0000 0050 4b05 0600 0000 0001 0001 004e

0000 0085 0000 0000 00

第一眼看过去以为是什么没见过的加密，查网上没有任何结果，将第一对数字拉到converter里面转码，经过一(xia)顿（J）分（B）析（an），发现在hex 转text的时候出现以下结果



PK的字眼怀疑是压缩包，由于内容不多，而且直接粘贴不行，直接新建文件在notepad手动输入十六进制。得到的文件为一个加密的压缩包。确认不是伪加密直接拿到Ziperello解密，得到密码



打开里面的文件即可的到flag，但是提交的时候提示说以提交过，这时才知道为什么要找目标QQ换flag，最终从目标QQ那里得到flag

hgame{zhe\_Sh1\_true\_F14g23333333333}

参考文章：

[1] <https://www.cnblogs.com/tk091/p/3410245.html>

Crypto

分数100：babyRSA

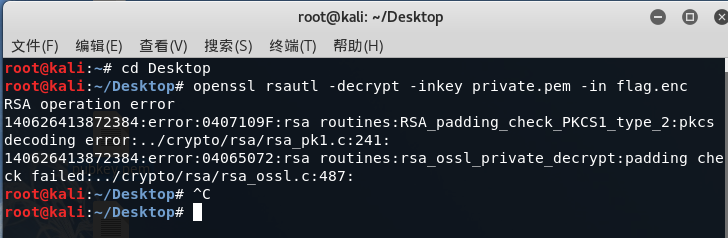
描述 你真的会用openssl了吗

hint: RSA的填充

URL：<http://p3xlhyup6.bkt.clouddn.com/babyRSA.zip>

下载压缩包得到3个文件（flag.enc，private.pem和pubkey.pem），放到kali linux中解密，在命令端输入

openssl rsautl -decrypt -inkey private.pem -in flag.enc，但是会报错



通过报错信息查找网上无果，后来题目出hint，通过RSA填充查到方法[1]

在命令端输入openssl rsautl -decrypt -inkey private.pem -in flag.enc –oaep 之后得到flag

hgame{OAEP\_i3\_safer%$#}

参考文章

[1] <http://heartsky.info/2017/01/13/SCTF-2016-RSA-writeup/>