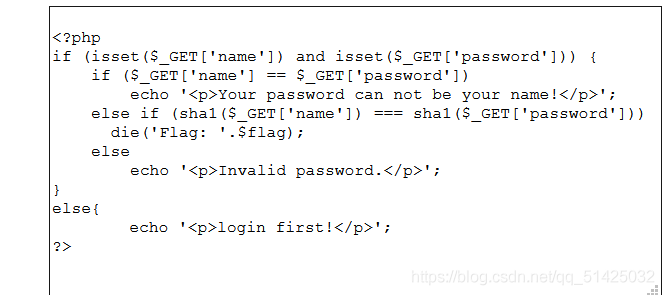
**Web**

**babyphp**

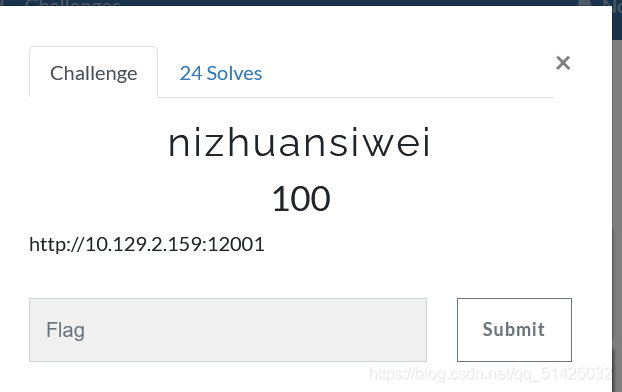
首先第一步由题目可知该题与PHP有关，随后进入网址​

第二步，进入网址打开源码进行代码审计​

可知想得到flag，必须要是SHA1()与SHA1（）中的值即为name和password经过sha1()函数后一样，经过查资料可得，要绕开sha1()函数，需使用数组类型的参数

第三步，随即在网址输入框中输入/challenge.php?name[]=a&password[]=b进行传参就可以绕过sha1函数得到flag

**逆转思维**

首先第一步观察题目“逆转思维”可知解题可能跟反向思维有关，随后进入网址​

第二步进入网址可得代码如下图​

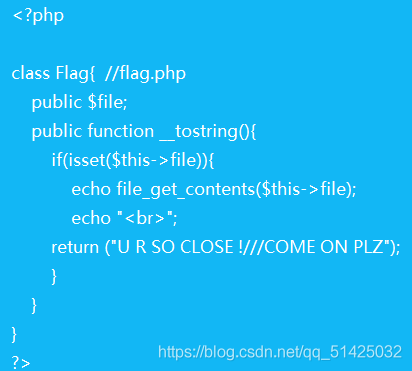
由题中的unserialize可知此题跟反序列化有关，这里附上序列化和反序列化的资料<https://blog.csdn.net/tree_ifconfig/article/details/82766587>，随后进行代码审计。

主要是3个参数text、password、file，file\_get\_contents($text,'r')==="5Y6f6aKY5pyq5pS55Yqo"，file\_get\_contents读取text文件的内容，恒等于5Y6f6aKY5pyq5pS55Yqo，text内容必须是5Y6f6aKY5pyq5pS55Yqo，password必须是反序列化。

第三步，代码审计结束后，得到足够的线索，进行第一次传参?text=data://text/plain,5Y6f6aKY5pyq5pS55Yqo，得到第一个payload如下图所示​

第四步，查看之前代码中的useless.php，这里运用了文件包含，这里附上链接<https://www.cnblogs.com/bin1121/p/12606894.html>，进行第二次传参查看该文件?text=data://text/plain,5Y6f6aKY5pyq5pS55Yqo  
&file=php://filter/read=convert.base64-encode/resource=useless.php，得到一长串字符，经观察推测为base64形式

第五步，将useless.php的内容去base64的网页进行解码，得到一下结果如下图

​

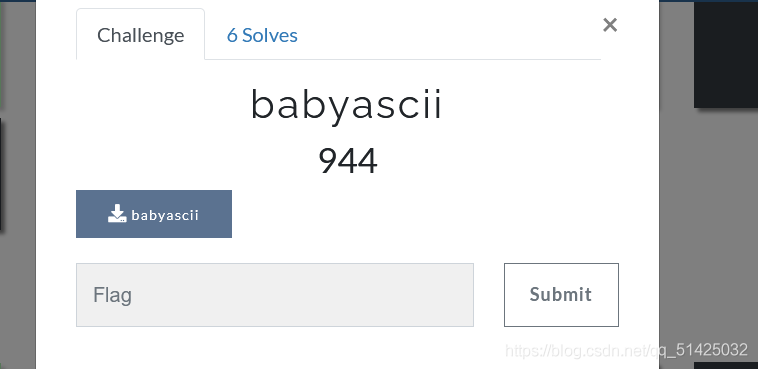
第六步，对整个文件进行反序列化（需用phpstudy搭建环境，打开网站根目录然后127.0.0.1/xx.php）结束反序列化后得到一串字符串O:4:"Flag":1:{s:4:"file";N;}，此为password的值

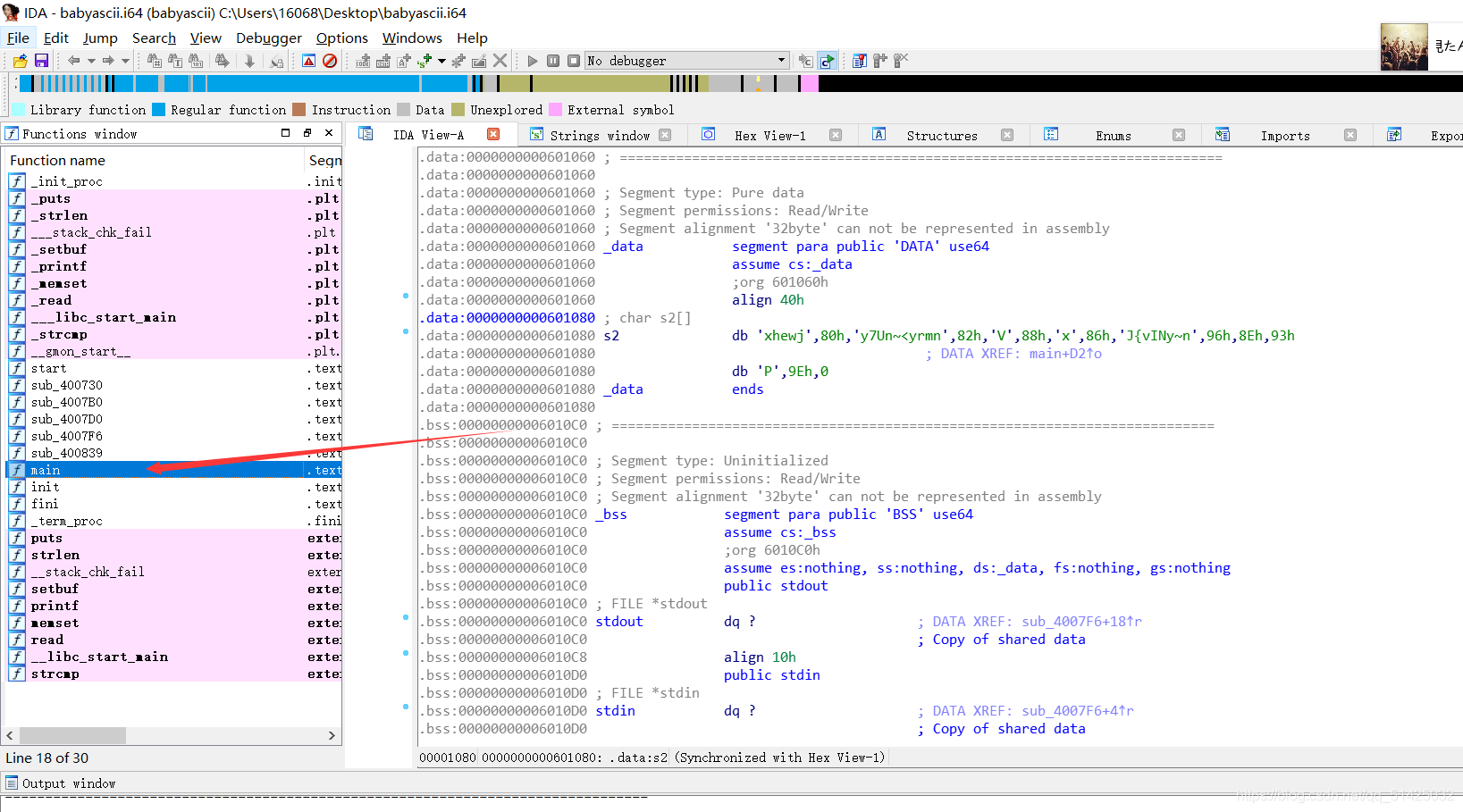
第七步继续代码审计​

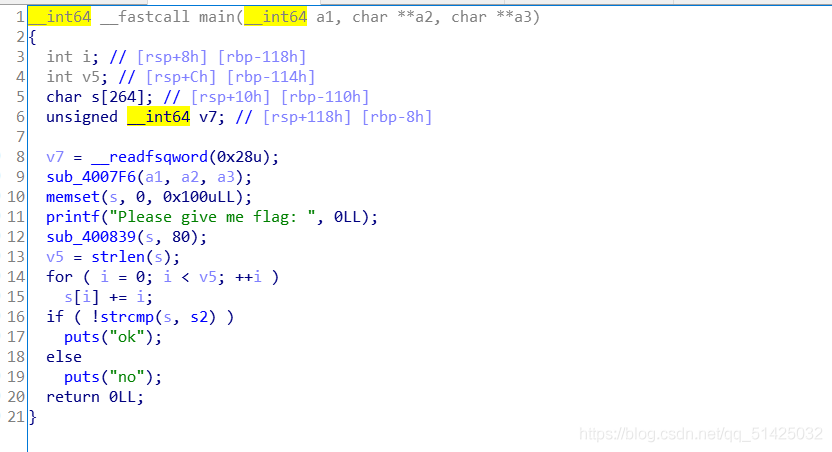
进行第三次传参?text=data:text/plain,5Y6f6aKY5pyq5pS55Yqo&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}对file的内容反序列化得到最终的payload

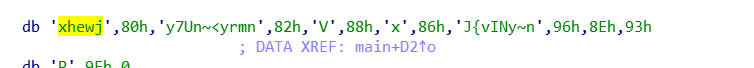
**re**

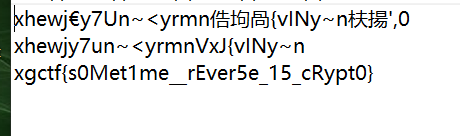
**babyascii**

先第一步观察题目可知与ASCII码有关，然后下载附件，并用IDA打开​

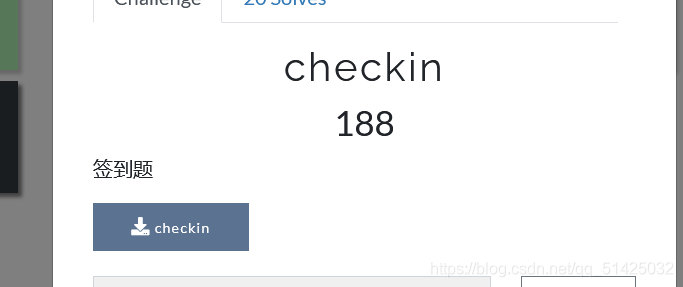
第二步用ida打开后观察左侧函数窗口，并找到主函数main，点击并进入伪代码​

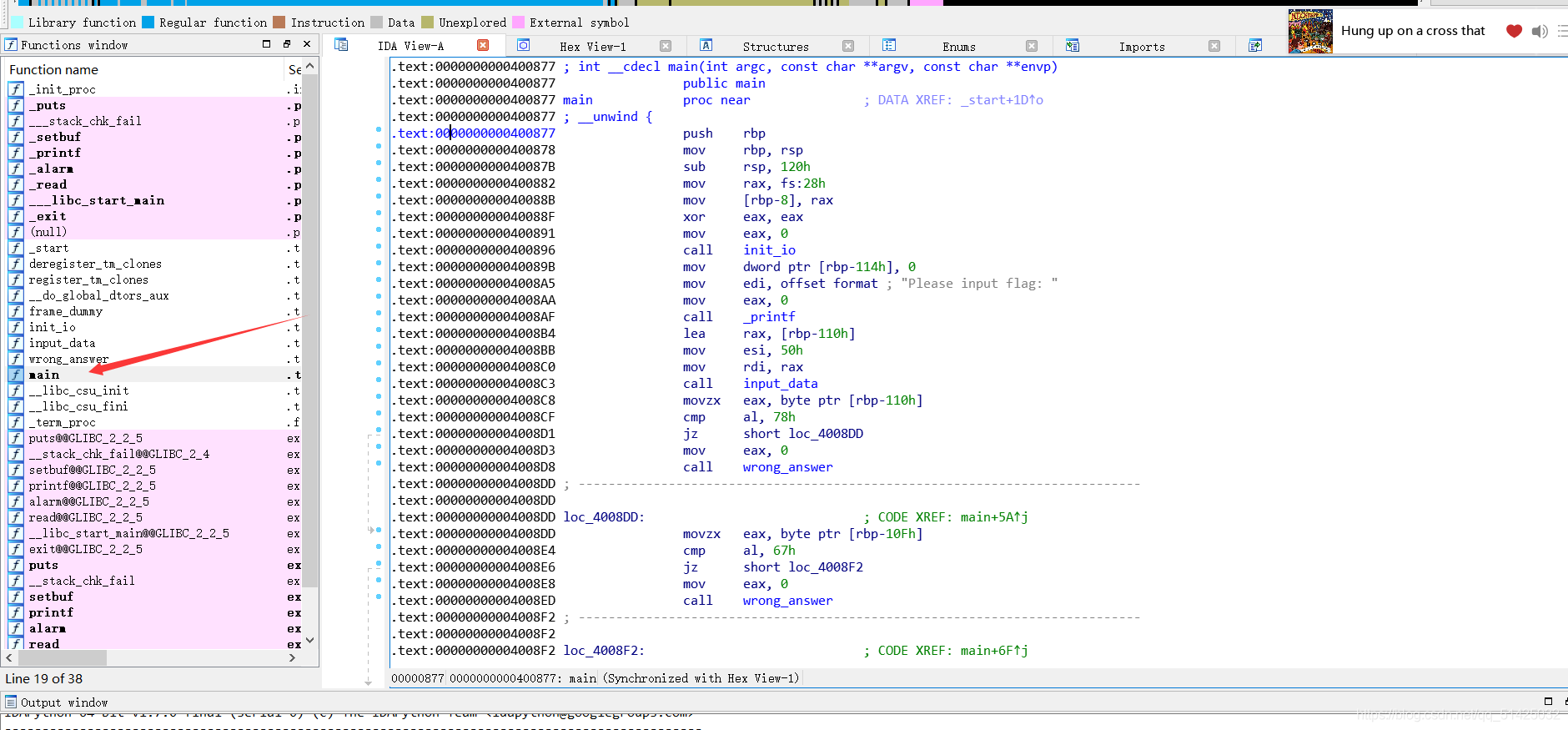
第三步进行代码审计，重点思考for循环和if语句可以知道此题答案与s和s2相关​

第四步点击s2可得s2的值如下图上方绿字​

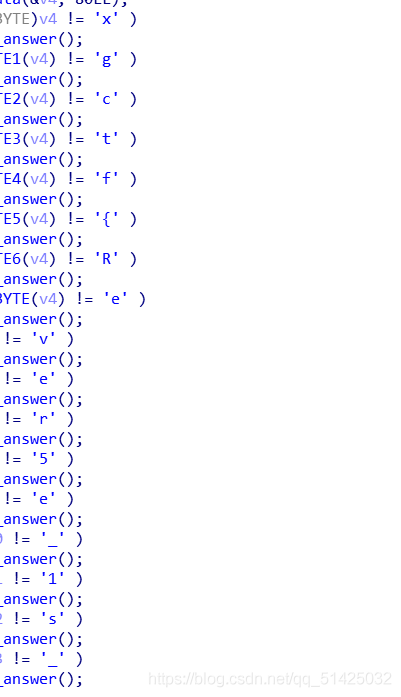
联想到题目的提示ASCII码，以及绿字中的80h诸如此类的字符，可以知道为16进制的ascii,再根据if和for的语句算法，编写脚本并代入绿字即可得到flag（也可跟本人一样艰辛手算）​

**checkin**

首先第一步，观察题目，发现并没提示，随后下载附件​

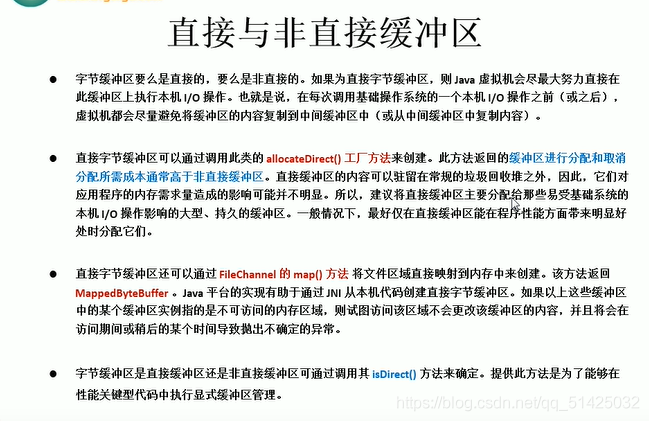
第二步，下载完成后用IDA打开，观察左侧函数窗口，并找到主函数main，点击此函数​

第三步，找到此函数后，观察伪代码，进行代码审计，发现每个If语句后均带有数字，猜测为ASCII码，随后点击每个数字并按下R键转为字符，得到flag

​

**pwn**

**babyshell**

依旧为一道签到题，首先第一步看题目并未给出提示，按照指示输入指令​

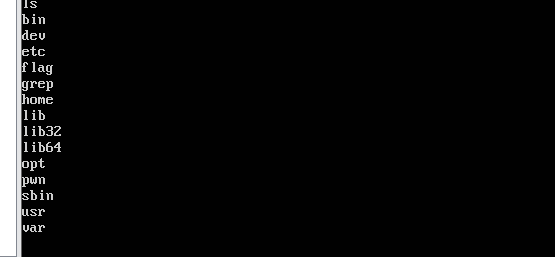
第二步输入后得到如下图

​

得到线索后，输入合适的Linux命令获取flag

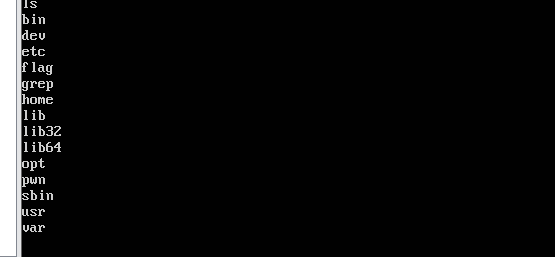
第三步输入ls查看目录，得到以下目录

​

直接使用cat flag或者ls flag会得到xgctf{fake\_flag}​

且经过本人多次翻看其他目录，出题人也在其他目录放了fake flag，所以建议采用find +其他参数寻找flag

第四步最终在usr/bin目录发现flag如下图

​

**babystackflow**

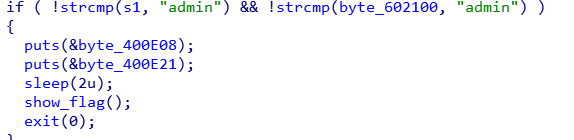
先第一步查看题目，无任何提示，直接把附件拖进IDA进行查看

​

第二步IDA内直接进入伪代码可得下图

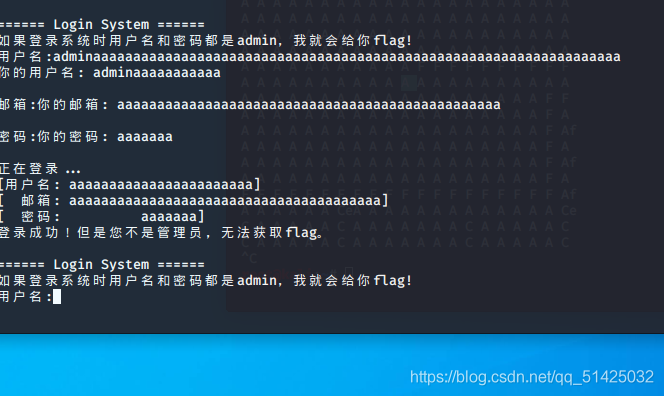
​

查看该函数的逻辑，观察是否存在栈溢出，发现个地方对输入的字符串长度没有检测，导致可以覆盖掉后面的内容即下图

​

可知进入这个if就出flag了或者可以直接跳进show\_flag的地址

第三步，直接把文件拖入虚拟机进行运行，首先尝试几个用户名和密码，如下图

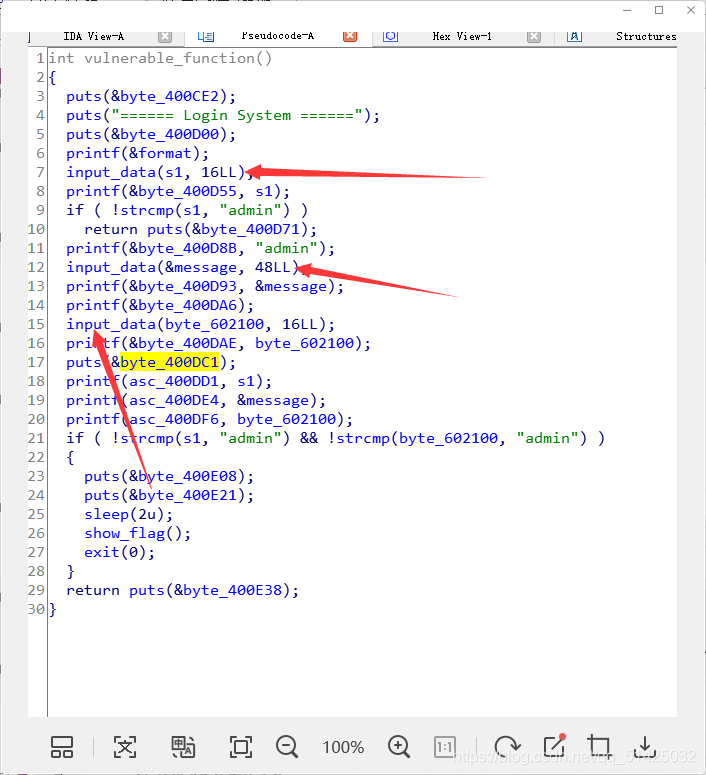
​

经过我个人测试如下图

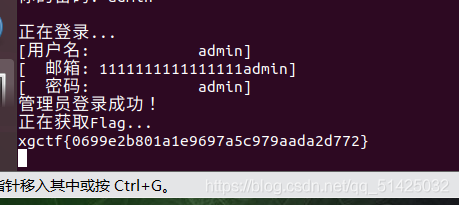
​

覆盖效果如上图

第四步又返回IDA查看，可得到用户名，邮箱，密码这三个变量的覆盖

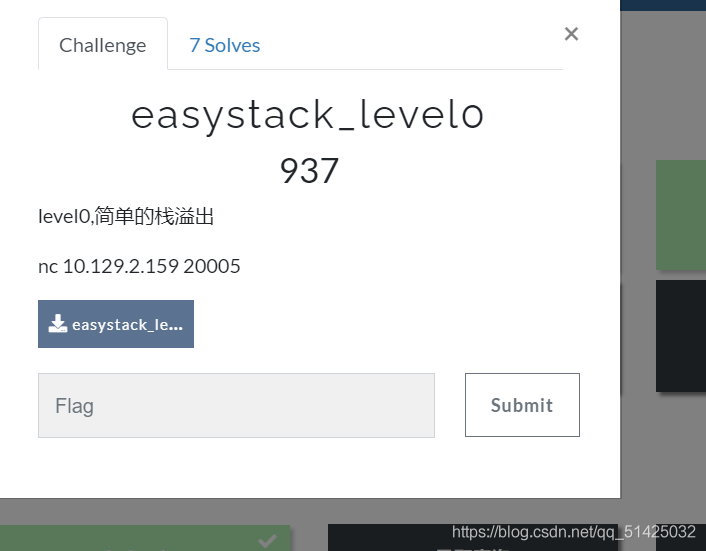
​

随后进行最后的尝试，即可得到flag

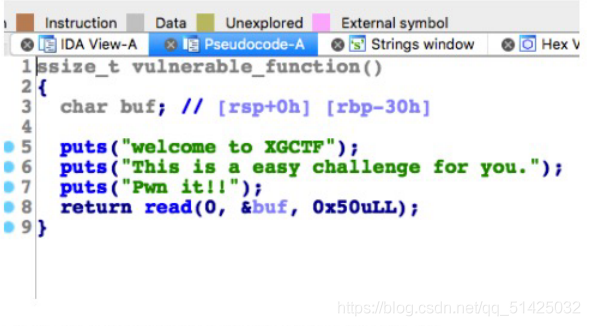
​

**easystack**

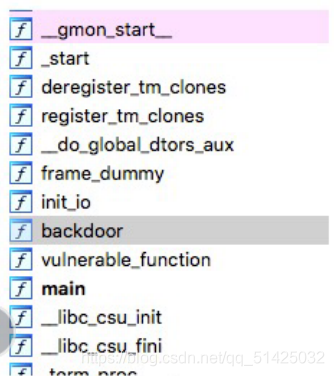
先第一步观察题目得知解题跟栈溢出有关，这里贴出相关知识<https://blog.csdn.net/yan_star/article/details/88937283>，

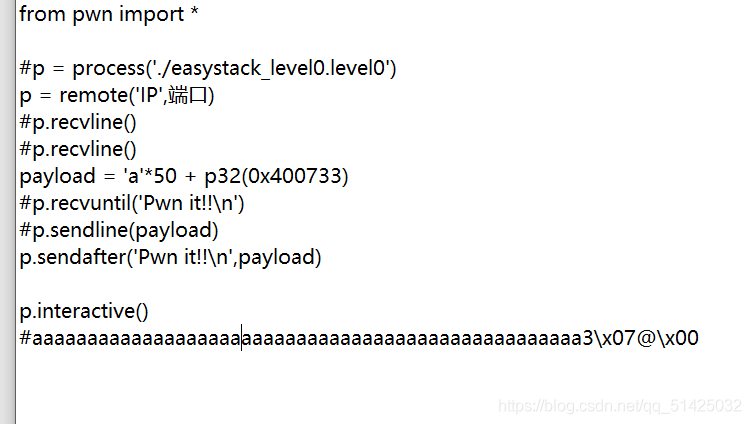
​

第二步进入Ida打开发现vulnerable\_function有栈溢出漏洞

​

第三步pwngdb调试出来buf在栈里面需要覆盖0x30大小的数据，又在ida里面有一个后面函数backdoor（），用ida找到其地址0x400733

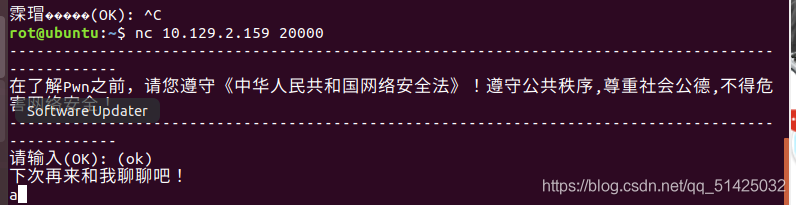
​

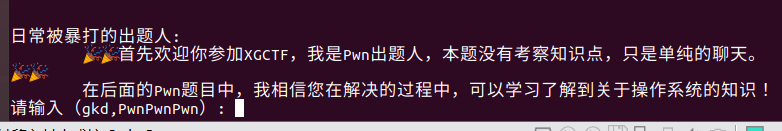
于是编写脚本如下图​

修改端口和Ip地址，在虚拟机里跑一下即可获得flag

**签到**

先第一步观察签到题的提示，直接按照提示进行安装相关环境及软件​

第二步安装完成后进入虚拟机的命令行，输入上图的命令，得到下图结果​

依次按照所给出的指令输入，得到一系列聊天日常如下图​

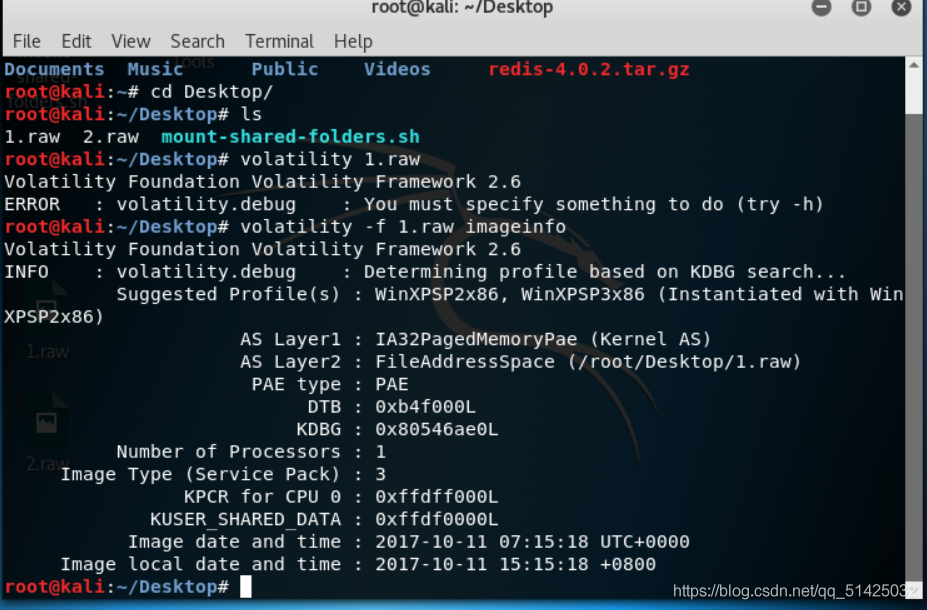
输入到最后即可得到flag

**misc**

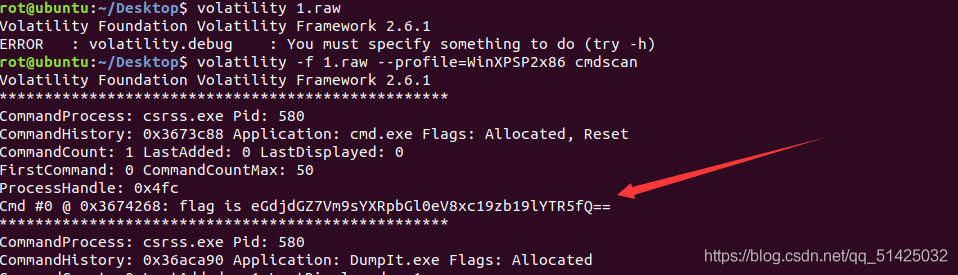
**内存取证**

首先第一步观察题目并未给出任何提示，随即下载附件​

第二步经查资料得，内存取证相关题目应用volatility工具进行，且应该在虚拟机进行，该工具kail自带（其他虚拟机系统需自行下载），在volatility内打开

● 下载后是一个.raw文件，raw意为“原材料，未加工的”。。  
● 进入.raw文件所在目录，输入命令判断该文件内存进程volatility -f 1.raw imageinfo​

第三步由上图可知，我们应在WinXPSP2x86进行寻找，此时需用到一个命令cmdscan

该命令cmdscan可获取曾经在cmd上输入过的内容，输入后可找到flag​

第四步得到flag后，发现并不是普通形式，为base64编码，进行解码后得到该题flag

**签到**

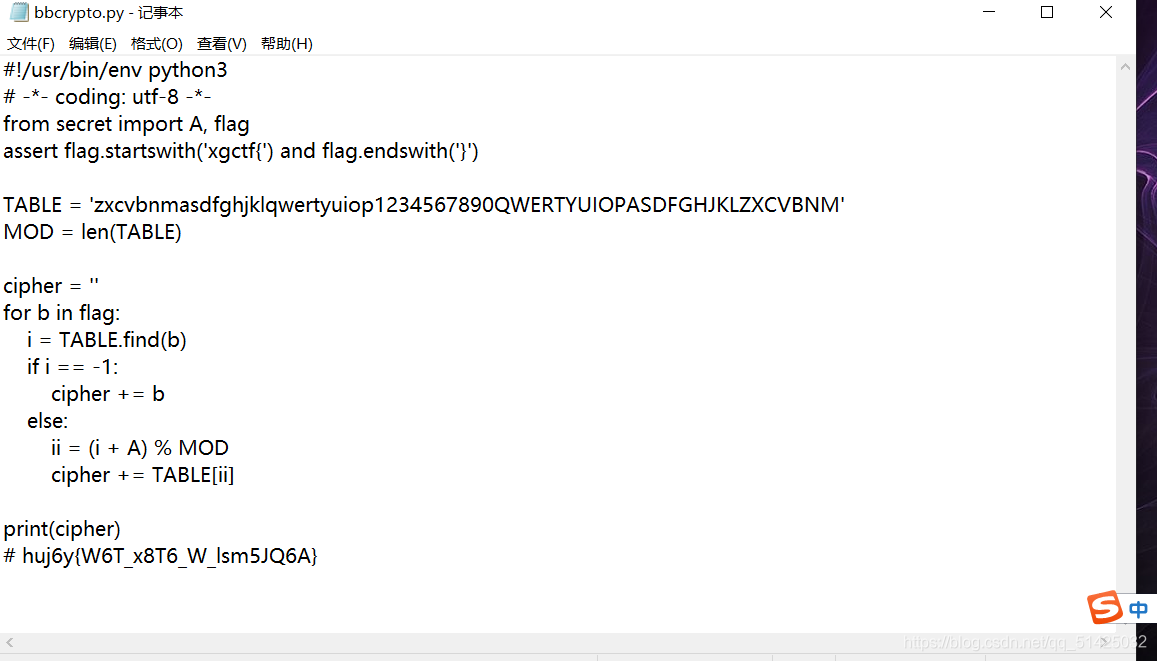
直接打开题目获得flag，复制提交即可

​

**crypto**

**bbcrypto**

首先第一步观察题目，可知解题也许跟凯撒密码有关，随后下载附件​

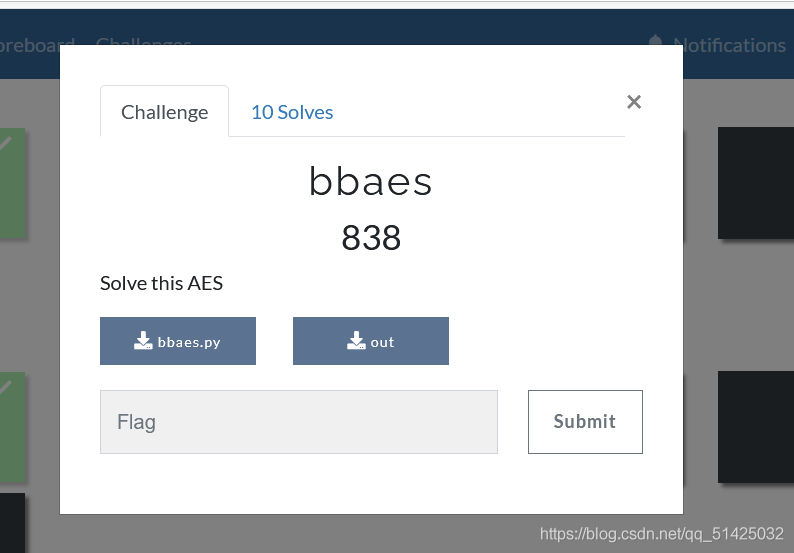
第二步打开附件后进行代码审计​

重点观察assert以后的代码，可知flag被分为两部分，且加密的密钥为TABLE的字符顺序，

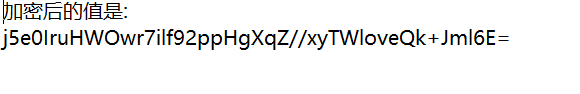
随后对for循环和If语句进行重点思考，可知取了flag里的每个字符进行加密，由下面加密后的huj6y{W6T\_x8T6\_W\_lsm5JQ6A}其中的huj6y联想到xgctf，并进行编写脚本解密（也可跟我一样艰辛手算）

第三步，经过解密后可得到flag xgctf{1t4\_Ju4t\_1\_bBCrYpt0}

**bbaes**

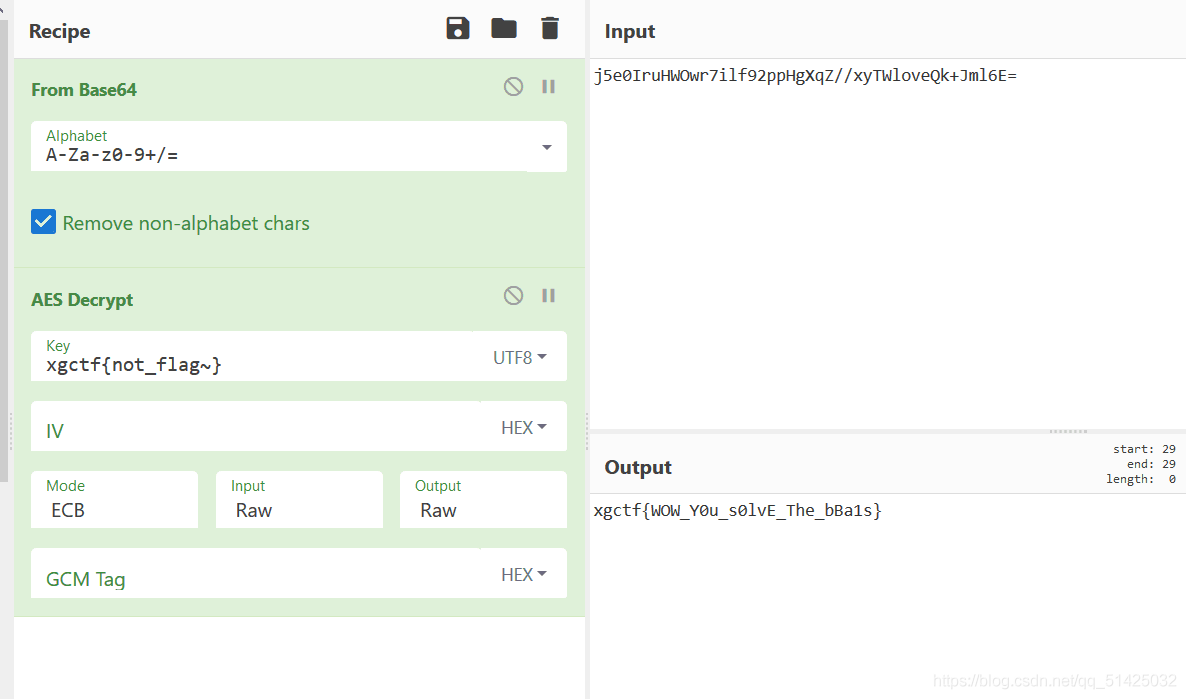
首先第一步观察题目，可知这是一道AES加密的题，随后下载题目附件及提示​

这里给出AES加密详细解释<https://blog.csdn.net/qq_28205153/article/details/55798628>

第二步打开out提示和bbaes文件观察​

​

进行代码审计后，可知我们已得到密钥xgctf{not\_flag~}和加密后的字符，随后进行解密

第三步，先对加密的后的字符进行base64解码，得到A-Za-z0-9+/=，再利用在线aes解密工具代入密钥即可得到flag​

刚入门时第一次打的比赛也是我学校的院赛，难度很简单供大家学习。