每次做完一道题目我都要说一句：pwnable.kr 牛逼！
一道密码题目
首先给了我们客户端和服务端的代码
我们可以知道用户输入id和pw后转换成 id-pw-cookie的形式然后用AES CBC模式加密后传送给服务器

服务器解密后取出id，pw，cookie，用于登录，通过形式是id相同且 sha256（id+cookie)=pw
也就是说 密码是sha256（id+cookie）
我们的目标是admin用户，即id=admin
但是不知道pw，怎么办呢？我们知道，pw = sha256（id+cookie）
id已知，知道了cookie就能知道pw了
但是cookie怎么解决?

参考
https://github.com/victor-li/pwnable.kr-write-ups/blob/master/crypto1.md
主要漏洞是，id和pw输入是没有长度限制的
首先可以id输入61个‘-’，pw不输入，这样加密字符串的前63个字符都是‘-’，接下来就是cookie的第一个字符。
接下来id输入63个’-‘+一个爆破字符，可以爆破出cookie的第一个值
以此类推能够算出所有的cookie
至于为什么要输入前面61个’-‘，因为可以测试出cookie的长度为49

```
from pwn import *
#context.log_level='debug'

def get_ciphertext(packet):
    ciphertext = ''
    io = remote('pwnable.kr',9006)
    io.recvuntil("Input your ID")
    io.sendline(packet)
    io.recvuntil("Input your PW")
    io.sendline('')
    io.recvuntil('(')
    ciphertext = io.recvline().split(')')[0]
    return ciphertext

cookie_len=49
pad_len = 64
cookie = ''
cipher_book = '1234567890abcdefghijklmnopqrstuvwxyz-_'
for i in range(1,cookie_len+1):
    packet = '-'*(pad_len-i-2)
    cipher = get_ciphertext(packet)[0:2*pad_len]
    #print(cipher)
    print("number %d"%(i))
    for j in cipher_book:
        packet = '-'*(pad_len-i-2)+'--'+cookie+j
        tmp = get_ciphertext(packet)[0:2*pad_len]
        if tmp== cipher:
            print('find!')
            print(packet)
            cookie+= j
            break
print(cookie)
```