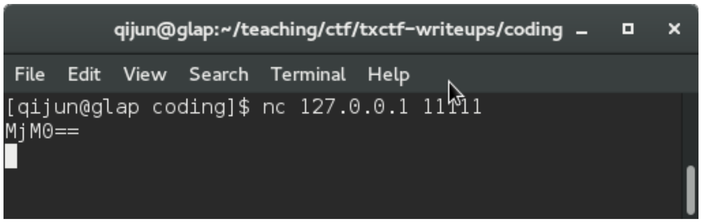
Hacker Level

Problem: If you answer 10000 problems, the flag is yours. I want the decimal version of these numbers.
Hint: Your pwn level is over 9000!
Given: nc 127.0.0.1 11111

Notes: Pwntools library will be used. To learn more about how to use and get pwntools visit this link.

Steps:
1) Connect to the server and see what the output is.
Note: before you start making your script is easier to connect using netcat and see the output.



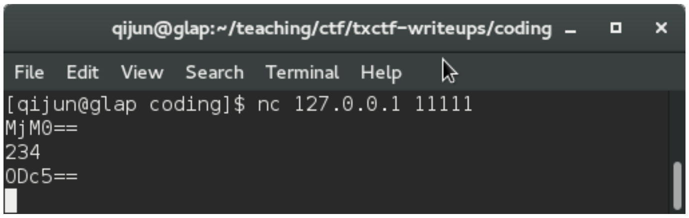
qijun@glap:~/teaching/ctf/txctf-writeups/coding
File Edit View Search Terminal Help

```
[qijun@glap coding]$ nc 127.0.0.1 11111
MjM0==
```

2) After connecting to the server, you can notice that the received information is in base64. Let's decode this base64 string that the server gave us and see what it is.

MjM0== is 234

3) Now we know that we are given a base64 encoded version of a decimal number. So, we type 234 and send it back to the server. The server then provides another base64 string.

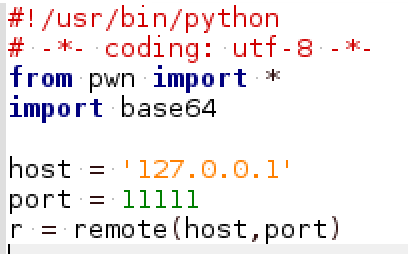


qijun@glap:~/teaching/ctf/txctf-writeups/coding
File Edit View Search Terminal Help

```
[qijun@glap coding]$ nc 127.0.0.1 11111
MjM0==
234
ODc5==
```

4) From the problem statement we know we have to give the server the correct decimal value for 10000 different problems. We can only do this with a script. This script should a) connect to the server, b) get info from the server, and c) send the data back to the server. This needs to be in a loop to go until the server gives you the flag.

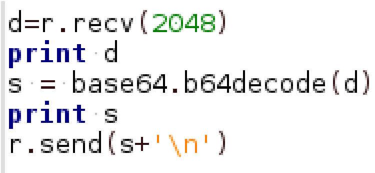
5) The short snippet below connects to the remote server at 127.0.0.1 at port 11111.



```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from pwn import *
import base64

host = '127.0.0.1'
port = 11111
r = remote(host,port)
```

6) Now, we receive data “d” from the server and decode it with base64 to a string “s”. Then, we send “s” back to the server. Note, base64.b64decode() is the function to decode a base64 string.



```
d=r.recv(2048)
print d
s = base64.b64decode(d)
print s
r.send(s+'\n')
```

7) Now we need to create a loop to get data from the server and send response back to the server automatically. We notice that the base64 strings from the server end with “==”. So, we use “==” to determine if we need to answer the problems or we receive a flag, assuming the flag does not have “==” in it.