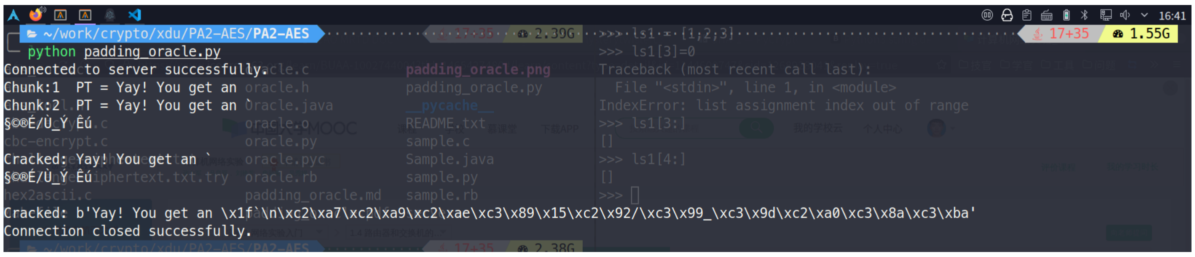
Padding_oracle 漏洞利用

原理: 详见seebug

实现:

```
import sys
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
from oracle import *

def fixed_xor(a,b):
    assert(len(a) == len(b))
    return "".join([chr(x^y) for (x,y) in zip(a,b)])

def padding_oracle_bug(ct):
    ct_list = [ct[i:i+AES.block_size] for i in range(0,len(ct),AES.block_size)]
    pt = ""

    for chunk_pos in range(len(ct_list)-1):
        # 每次ct_work的前一块为iv,后一块为待解密块
        ct_work = [ct_list[chunk_pos], ct_list[chunk_pos+1]]
        iv_work = [0] * AES.block_size
        median_work = [0] * AES.block_size
        for iv_byte_pos in range(AES.block_size-1,-1,-1):
            # iv_work[:iv_byte_pos-1] = [0]*(iv_byte_pos-1)
            iv_work[:iv_byte_pos] = [0]*(iv_byte_pos)
            iv_work[iv_byte_pos+1:] = [x^(AES.block_size-iv_byte_pos) for x in
median_work[iv_byte_pos+1:]]
            for iv_byte_value in range(0x00,0x100):
                iv_work[iv_byte_pos] = iv_byte_value
                rc = Oracle_Send(iv_work+ct_list[chunk_pos+1],2)
                if rc == 49:
                    median_work[iv_byte_pos] =
iv_work[iv_byte_pos]^(AES.block_size-iv_byte_pos)
                    break
        pt += fixed_xor(ct_work[chunk_pos], median_work)
        print("Chunk:{}  PT = {}".format(chunk_pos+1, pt))
    return pt

if __name__ == "__main__":
    with open(sys.path[0]+"/challenge-ciphertext.txt") as cipherfile:
        data = cipherfile.read().strip()
        ct = [(int(data[i:i+2],16)) for i in range(0, len(data), 2)]
    Oracle_Connect()
    pt = padding_oracle_bug(ct)
    print("\nCracked:",pt.strip())
    print("\nCracked:",pt.encode())
    Oracle_Disconnect()
```

运行结果:



```
~/work/crypto/xdu/PA2-AES/PA2-AES
python padding_oracle.py
Connected to server successfully.
Chunk:1  PT = Yay! You get an
Chunk:2  PT = Yay! You get an `
§©®É/Ù_Ý Êú

Cracked: Yay! You get an `
§©®É/Ù_Ý Êú

Cracked: b'Yay! You get an \x1f`\n\xc2\xa7\xc2\xa9\xc2\xae\xc3\x89\x15\xc2\x92/\xc3\x99_\xc3\x9d\xc2\xa0\xc3\x8a\xc3\xba'
Connection closed successfully.
```

(运行结果奇奇怪怪，实在找不到问题出在哪)