Network Security WS17/18
Challenge 05

Exercise 5 is hosted at netsec.net.in.tum.de at port 20005. Bob operates a simple hand-written FTP server there. The FTP service supports the following commands: `"SEND ENCRYPTED DATA"` and `"SEND DATA"`. We provide you with an example client, `alice.py`.
Unfortunately, you don't have the key to decrypt the encrypted data. Therefore, you should try to send the command `"SEND DATA"`.
Bob doesn't want unauthorized persons to get the data. Only people who know the symmetric key should be allowed to get it. Therefore, Bob tried to patch the server such that it will only give away the data encrypted.

```
if len(cmd) != len("SEND ENCRYPTED DATA"):
            client_writer.write("Bob does not allow commands of length {}\n"
                        .format(len(cmd)).encode())
            return
```

Details about the code

For encryption, Bob uses the block cipher AES[1]. You have probably learned that block ciphers operate on data blocks of fixed length (for AES: 16 Bytes). If the length of the data is not a multiple of the blocksize, the data needs to be *padded*. Bob invented his own padding scheme: He simply adds underscores ('_').

```
if (len(plaintext) % 16 != 0):
    plaintext += b'_' * (16 - len(plaintext) % 16)
```

The encryption function looks as follows: Bob chooses a new, fresh IV. Then, Bob adds the padding to the data and encrypts it blockwise with AES.

```
def encrypt(plaintext):
    iv=os.urandom(AES.block_size)
    #add padding
    if (len(plaintext) % 16 != 0):
        plaintext += b'_' * (16 - len(plaintext) % 16)
    cipher = AES.new(encryption_key, AES.MODE_CBC, iv)
    ciphertext = cipher.encrypt(plaintext)
    return hexlify(iv) + b"," + hexlify(ciphertext) + b"\n"
```

There is also code to remove the padding again.

```
cmd = cmd.replace('_', '')
```

Can you get the data? Hint: have a look at the source code.

Note: In this hacking task, you do not need any processing of crypto in your client. There are two main python crypto libraries which you can use in subsequent hacking tasks. One is sufficient. To run `Bob.py`, on debian/ubuntu, you need to install `python3-crypto`. `Bob_alt.py` contains the same code, but also using library `python3-cryptography`, which has more features. You could change the main encryption function from one framework to the other if you like. The file also contains examples of encryption and decryption with the two libraries. Check the homepages of the libraries `https://pypi.python.org/pypi/pycrypto` and `https://cryptography.io/en/latest/` for documentation and installation guidelines for other systems.

[1] As you learned, AES is pretty secure; do not try to break the crypto.