Network Security WS15/16
Challenge 04

Exercise 4 is hosted at netsec.net.in.tum.de at port 20004. 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 heard 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. You will learn in the second week of November why this is important. Then, Bob adds the padding to the data and encrypts it blockwise with AES.

```
def encrypt(plaintext):
    iv = Random.new().read(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: to run `bob.py`, on debian/ubuntu, you need to install `python3-crypto`.

[1] We will learn about encryption in the following weeks. For this exercise, everything you need to know is in the description. You don't know the encryption key, don't try to break the crypto.