

### Results:

1. When credentials are wrong, Arduino sends to python "CONNECTION NOT ALLOWED"

Home Page - Select o x

security - Jupyter Not x

myCourseVille x

+  
localhost:8889/notebooks/security.ipynb#

jupyter security (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O

In [8]:

# while True:  
#Request for authentication  
s.sendto("Request for connection".encode('utf-8'), (HOST, PORT))  
print("Python to Arduino: Request for connection")  
# #Send Authentication  
data, addr = s.recvfrom(PORT)  
print("Arduino to Python(iv): ", data)  
  
#Send username in cleartext and password in encrypted text separated by comma  
username=bytes("Chimi", 'ascii')  
password="Happyme1!"  
plaintext = bytes(password, 'ascii') # did I convert these bytes correctly f  
passkey = bytes("11111111111111", 'ascii') #have I converted the passkey f  
# iv = b64encode(os.urandom(16)).decode('utf-8')  
IV = data  
IV=IV[:len(IV)-1] # have I converted the IV from the example code correctly  
encrypted\_bytes = encrypt(plaintext, passkey, IV)  
print("Python to Arduino (Cipher text): ", encrypted\_bytes)  
# s.sendto(encrypted\_bytes, (HOST, PORT))  
# print(type(encrypted\_bytes))  
# s.sendto(bytes(username+";"+str(encrypted\_bytes), "ascii"), (HOST, PORT))  
# s.sendto((username+";"+str(encrypted\_bytes)).encode('utf-8'), (HOST, PORT))  
# print(type((username+";"+str(encrypted\_bytes)).encode('utf-8')))  
s.sendto(username+b";"+encrypted\_bytes, (HOST, PORT))  
message, addr = s.recvfrom(PORT)  
message, addr = s.recvfrom(PORT)  
print("Arduino to python (Acknowledgement): ", str(message.decode('utf-8')))  
  
for i in range(3):  
s.sendto("BLINK\_LED\_A AND BLINK\_ON\_LED\_B".encode('utf-8'), (HOST, PORT))  
print("Python to Arduino: BLINK\_LED\_A AND BLINK\_ON\_LED\_B")  
  
Python to Arduino: Request for connection  
Arduino to Python(iv): b'\xbcb\xbc\xbc\xbc\xbc\xbc\xbc\xbc\xbc\xbc\xbc\xbc\xbc\xbd1'  
Python to Arduino (Cipher text): b'dgVvc6h5yu0rPQT2LGelAhG0cKMDRCwGPqGbp+pVZ4c=  
=  
Arduino to python (Acknowledgement): CONNECTION NOT ALLOWED  
Python to Arduino: BLINK\_LED\_A AND BLINK\_ON\_LED\_B  
Python to Arduino: BLINK\_LED\_A AND BLINK\_ON\_LED\_B  
Python to Arduino: BLINK\_LED\_A AND BLINK\_ON\_LED\_B

COM6  
14:49:21.361 ->  
14:49:21.361 -> Python to Arduino (Authentication Request):Request for connection  
14:49:21.414 -> Arduino to Python (iv):????????????????  
14:49:21.461 -> Waiting.... 1  
14:49:21.461 ->  
14:49:21.461 -> Python to Arduino (Authentication Request):Request for connection  
14:49:21.554 -> Arduino to Python (iv):????????????????  
14:49:21.614 -> Python to Arduino (Username):Chimi  
14:49:21.614 -> Python to Arduino (Password):dgVvc6h5yu0rPQT2LGelAhG0cKMDRCwGPqGbp+pVZ4c=  
14:49:21.684 -> Decrypted text:Happyme1!  
14:49:21.719 -> Arduino to python :CONNECTION NOT ALLOWED  
14:49:21.752 -> 0  
14:49:21.786 -> 0  
14:49:21.786 -> 0  
14:49:21.786 -> 0  
14:49:21.786 -> 30  
14:49:21.786 -> 30  
14:49:21.786 -> 30  
14:49:21.786 -> 0  
14:49:21.786 -> 0  
14:49:21.786 -> 0  
14:49:26.676 ->  
14:49:26.676 -> Backtrace: 0x4008c434:0x3fffb1eb0 0x4008c665:0x3fffbled0 0x400d7b50:0x3fffb1ef0 0x  
14:49:26.826 ->  
14:49:26.826 -> Rebooting...  
14:49:26.826 -> I?Q?;??s?;HCfIbR????????????????????????????????????????????????????????

Autoscroll Show timestamp Newline 9600 baud Clear output



(B) If the user is "Alice", only led A blinks 3 times.

The screenshot displays a Jupyter Notebook titled "security - Jupyter Notebooks" running on a local server at localhost:8889. The notebook contains a Python script that simulates a secure communication protocol between a host (Python) and an Arduino. The script uses a combination of symmetric encryption (XOR) and a one-time pad (OTP) for confidentiality.

```
In [8]: 
1 # while True:
2 #Request for authentication
3 s.sendto("Request for connection".encode('utf-8'), (HOST, PORT))
4 print("Python to Arduino: Request for connection")
5 #Send Authentication
6 data, addr = s.recvfrom(PORT)
7 print("Arduino to Python(iv): ", data)
8
9 #Send username in cleartext and password in encrypted text separated by comma
10 username=bytes("Alice", 'ascii')
11 password="Happyme11!"
12 plaintext = bytes(password, 'ascii') # did I convert these bytes correctly f
13 passkey = bytes("1111111111111111", 'ascii') #have I converted the passkey f
14 # iv = b64encode(os.urandom(16)).decode('utf-8')
15 IV = data
16 IV=IV[:len(IV)-1] # have I converted the IV from the example code correctly
17 encrypted_bytes = encrypt(plaintext, passkey, IV)
18 print("Python to Arduino (Cipher text): ", encrypted_bytes)
19 # s.sendto(encrypted_bytes, (HOST, PORT))
20 # print(type(encrypted_bytes))
21 # s.sendto(bytes(username+";"+str(encrypted_bytes), "ascii"), (HOST, PORT))
22 # s.sendto((username+";"+str(encrypted_bytes)).encode('utf-8'), (HOST, PORT))
23 # print(type((username+";"+str(encrypted_bytes)).encode('utf-8')))
24 s.sendto(username+b";"+encrypted_bytes, (HOST, PORT))
25 message, addr = s.recvfrom(PORT)
26 message, addr = s.recvfrom(PORT)
27 print("Arduino to python (Acknowledgement): ", str(message.decode('utf-8')))
```

The output of the script shows the following sequence of events:

- Python sends a request for connection to the Arduino.
- The Arduino responds with its IP address and port number.
- Python sends the username "Alice" and the encrypted password "Happyme11!".
- The Arduino responds with the decrypted password "Happyme11!".
- Python sends the cipher text "b'rzgibxps3S0X08QbeBPhkL4SYFJDSPjA5zDVVArC+kM'" to the Arduino.
- The Arduino responds with the acknowledgement "CONNECTED".
- Python sends the command "BLINK\_LED\_A AND BLINK\_ON\_LED\_B" to the Arduino.
- The Arduino responds with the acknowledgement "BLINK\_LED\_A AND BLINK\_ON\_LED\_B".