**Week 10:**

**Write a python program for client-server based intruder detection system using mqtt application layer protocol.**

**Steps to install MQTT protocol and Mosquitto - open-source MQTT broker**

**1. sudo apt install python3-paho-mqtt**

* Installs the **Paho MQTT Python client library** for Python 3.
* This allows Python scripts to connect to MQTT brokers, publish messages, and subscribe to topics using the **MQTT protocol**

**2. sudo apt install mosquito**

* Installs **Mosquitto**, an open-source **MQTT broker**.
* A broker acts as the central hub for MQTT clients — it receives messages from publishers and routes them to subscribers.

**3. sudo systemctl start mosquito**

* Starts the **Mosquitto service** using systemctl
* This means your MQTT broker is now **running in the background**, ready to accept MQTT connections.

**Program:**

**# client.py**

import time

import RPi.GPIO as GPIO

import paho.mqtt.client as mqtt

**# GPIO Setup**

PIR\_SENSOR\_PIN = 17

LED\_PIN = 27

GPIO.setmode(GPIO.BCM)

GPIO.setup(PIR\_SENSOR\_PIN, GPIO.IN)

GPIO.setup(LED\_PIN, GPIO.OUT)

**# MQTT Setup**

BROKER = "localhost"

PORT = 1883

TOPIC = "intruder/alert"

client = mqtt.Client()

client.connect(BROKER, PORT, 60)

try:

print("Motion detection client started...")

while True:

if GPIO.input(PIR\_SENSOR\_PIN):

print("Motion detected!")

GPIO.output(LED\_PIN, GPIO.HIGH) # Turn on LED

client.publish(TOPIC, "Intruder detected!")

time.sleep(1) # Debounce delay

else:

GPIO.output(LED\_PIN, GPIO.LOW)

time.sleep(0.1)

except KeyboardInterrupt:

print("Exiting program...")

finally:

GPIO.cleanup()

**# server.py**

import paho.mqtt.client as mqtt

from datetime import datetime

**# MQTT Setup**

BROKER = "localhost"

PORT = 1883

TOPIC = "intruder/alert"

def log\_status(message):

timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

print(f"[{timestamp}] {message}")

**# Called when the client connects to the broker**

def on\_connect(client, userdata, flags, rc):

if rc == 0:

log\_status("✅ Connected to MQTT Broker successfully.")

client.subscribe(TOPIC)

log\_status(f"📡 Subscribed to topic: '{TOPIC}'")

else:

log\_status(f"❌ Failed to connect, return code {rc}")

**# Called when a message is received**

def on\_message(client, userdata, msg):

payload = msg.payload.decode()

log\_status(f"🚨 ALERT RECEIVED: {payload}")

**# Set up client**

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

log\_status("🚀 Starting intruder detection server...")

client.connect(BROKER, PORT, 60)

**# Blocking loop**

client.loop\_forever()

**Execution:**

**Open two terminals 🡪 execute client and server code separately on two different terminals.**

**> python3 server.py**

**server output:**

[2025-04-07 07:31:59] 🚨 ALERT RECEIVED: Intruder detected!  
[2025-04-07 07:32:00] 🚨 ALERT RECEIVED: Intruder detected!  
[2025-04-07 07:32:02] 🚨 ALERT RECEIVED: Intruder detected!  
[2025-04-07 07:32:51] 🚨 ALERT RECEIVED: Intruder detected!

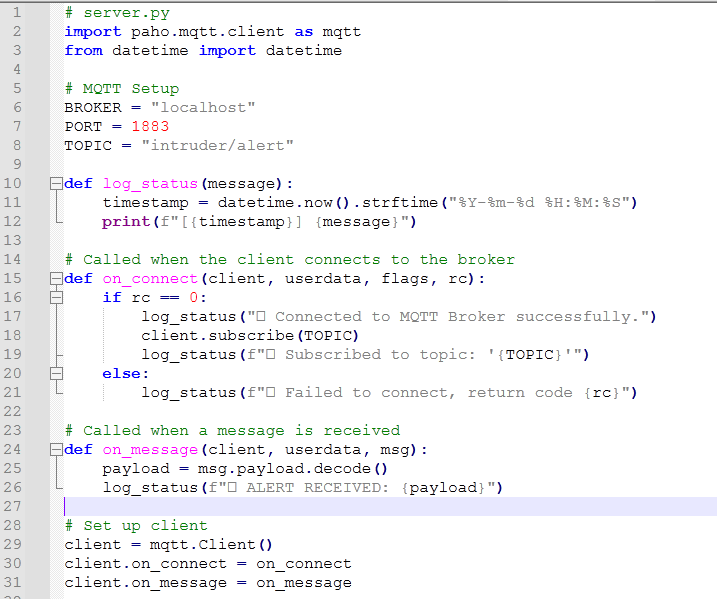
[2025-04-07 07:32:52] 🚨 ALERT RECEIVED: Intruder detected!  
[2025-04-07 07:34:02] 🚨 ALERT RECEIVED: Intruder detected!

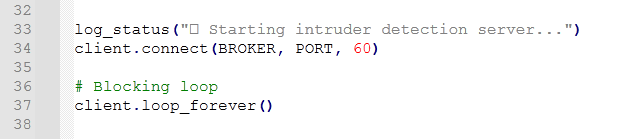
**>  python3 client.py**

**client output:**

Motion detection client started...  
Motion detected!  
Motion detected!  
Motion detected!  
Motion detected!

**server.py:**





**client.py:**

