EC-Council

codered
FROM EC-COUNCIL

CHAPTER 13

IOT AND OT SECURITY

CERTIFIED CYBERSECURITY TECHNICIAN

INDEX

LAB SCENARIO

IoT devices are vastly different from each other, and the security of devices relies on their type and model. With no or inadequate focus on IoT device security by manufacturers, the security measures used for IoT devices are often inadequate. Therefore, an organization should focus on securing IoT devices and countering the attack scenarios in IoT-enabled environments.
An adversary uses a compromised IoT device as an entry point to a network and performs a lateral movement attack. For example, a compromised smart printer can infect other systems and devices connected to the same network. A compromised router can spread malware to all the IoT devices connected to it. Hence, a security professional must focus on implementing IoT device security to prevent the devices from unauthorized access and data theft.

LAB OBJECTIVE

The objective of this lab is to provide expert knowledge in securing IoT and OT devices. This includes knowledge of the following tasks:

- Implementation of secure IoT device communication using TLS/SSL

OVERVIEW OF IOT AND OT SECURITY

To secure an IoT network and router, user should map and monitor all devices, apply network segmentation, ensure a secure network architecture, use routers with in-built firewalls, and disable unnecessary services such as Universal Plug and Play (UPnP). This helps in restricting the attacker from accessing other parts of the network and performing targeted attacks.
IT/OT convergence is being widely adopted in industries such as traffic control systems, power plants, and manufacturing companies. These IT/OT systems are often targeted by attackers to discover the underlying vulnerabilities and indulge in cyber-attacks. Based on the Purdue model, an IT/OT environment is divided into several levels, and each level must be secured with proper security measures.

LAB TASKS

A cyber security professional or a security professional use numerous tools and techniques to configure secure communication in IoT devices. The recommended labs that will assist you in learning the implementation of security controls in the IoT device communication include:

01 **Secure IoT Device Communication using TLS/ SSL**

Note: Turn on PfSense Firewall virtual machine and keep it running throughout the lab exercises.

EXERCISE 1: **SECURE IOT DEVICE COMMUNICATION USING TLS/SSL**

Encrypted communication over TLS/SSL is the key to securing IoT Device Communication.

LAB SCENARIO

As an ethical hacker or pentester or IT administrator, you should have sound knowledge of the protocols and their usages that can be implemented to create practical solutions to real-world problems. SSH is the protocol that helps to access remote control of the systems.

LAB OBJECTIVE

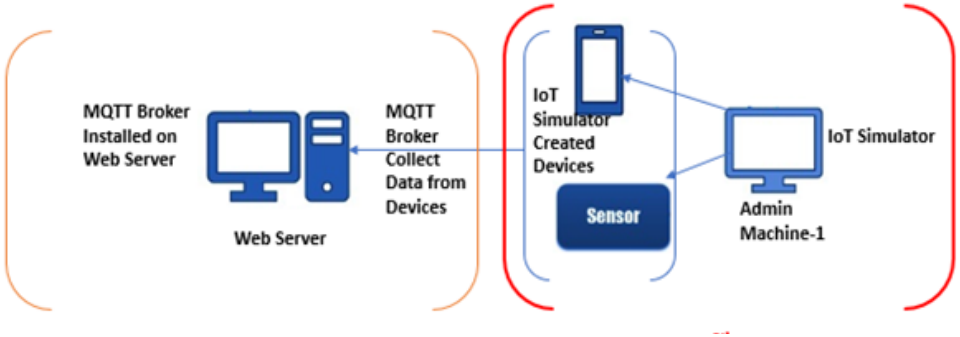
The objective of this lab is to demonstrate how to secure Internet of things (IoT) device communication using the Bevywise message queuing telemetry transport (MQTT) Broker and Simulator. This tool demonstrates the use of IoT devices over the virtual network. In this lab, you will learn to:

- Install and configure the Bevywise MQTT Broker.
- Implement transport layer security (TLS)/secure sockets layer (SSL) to secure IoT communication.

OVERVIEW OF BEVYWISE IOT SIMULATOR

MQTT is a lightweight messaging protocol that uses a publish/subscribe communication pattern. Because the protocol is meant for devices with a low-bandwidth, it is considered ideal for machine-to-machine (M2M) communication or IoT applications. We can create virtual IoT devices over the virtual network using the Bevywise IoT simulator on the client side and communicate these devices to the server using the MQTT Broker web interface. This interface collects data and displays the status and messages of devices connected over the network.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

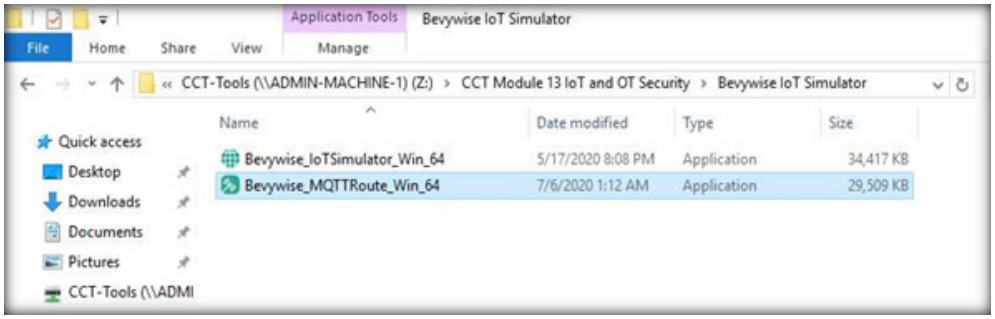


Note: Ensure that the **PfSense Firewall** virtual machine is running.

1. Turn on **Admin Machine-1** and **Web Server** virtual machines.
2. Switch to **Web Server** and Log in with the credentials **Administrator** and **admin@123.**

   Note: If the network screen appears, click **Yes.**
3. Navigate to the **Z:\CCT Module 13 IoT and OT Security\Bevywise IoT Simulator** folder and double-click on the **Bevywise_MQTTRoute_Win_64.exe** file.

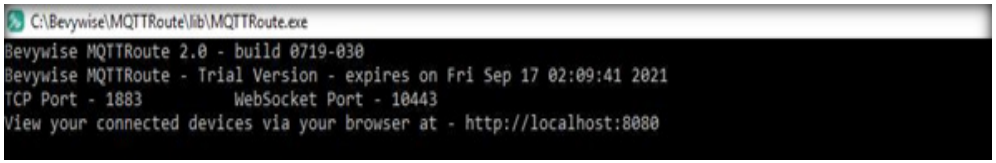
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



4. The **Open File - Security Warning** popup appears. Click **Run.**
5. The **Setup – MQTTRoute 2.0** window opens. Select **I accept the agreement** and click on Next.
6. The **Select Destination Location** page appears, without making any changes to the default installation location, click on **Next.**
7. In the next window, click **Install** to complete the installation process.
8. The installation completes; now, click on **Finish.** Ensure that **Launch Bevywise_MQTTRoute_Win_64** is checked.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

9. Now, the MQTTRoute will be executed, and the command prompt will appear. You can see that the **TCP** port using **1883.**

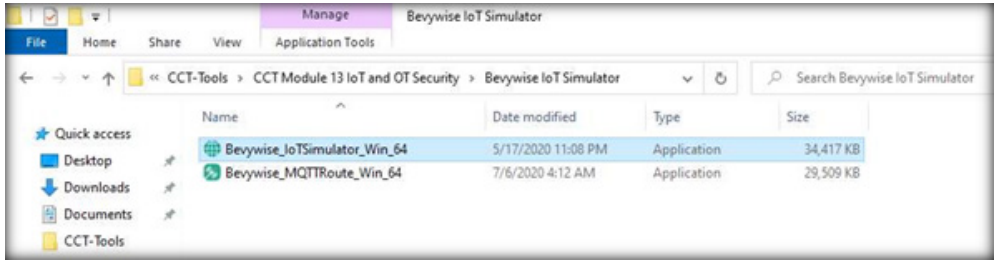


EXERCISE 1:
SECURE IOT DEVICE
COMMUNICATION
USING TLS/SSL

10. We have installed MQTT Broker successfully and leave the Bevywise MQTT **running.**
11. To create IoT devices, we must install the **IoT simulator** on the client machine.
12. Switch to the **Admin Machine-1** virtual machine.
13. Log in with the credentials **Admin** and **admin@123.**

    Note: If the network screen appears, click Yes.
14. Navigate to the **Z:\CCT-Tools\CCT Module 13 IoT and OT Security\Bevywise IoT Simulator** folder and double-click on the **Bevywise_IoTSimulator_Win_64.exe** file.

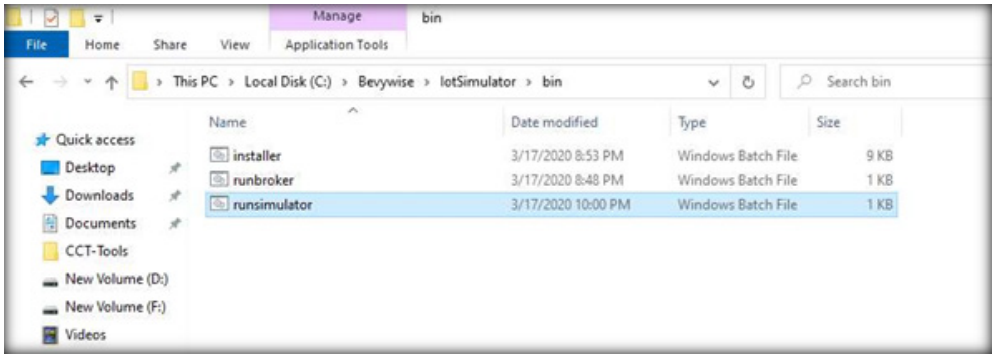
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



15. The **User Account Control** popup appears. Click on **Yes.**
16. The **Setup-IoTSimulator 2.1** setup wizard opens. Select **I accept the agreement** and click on Next to continue.
17. Do not change the default destination; then, click on **Next.**
18. The **Ready to Install** screen appears, click on **Install**
19. Click on **Finish** to complete the installation process.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

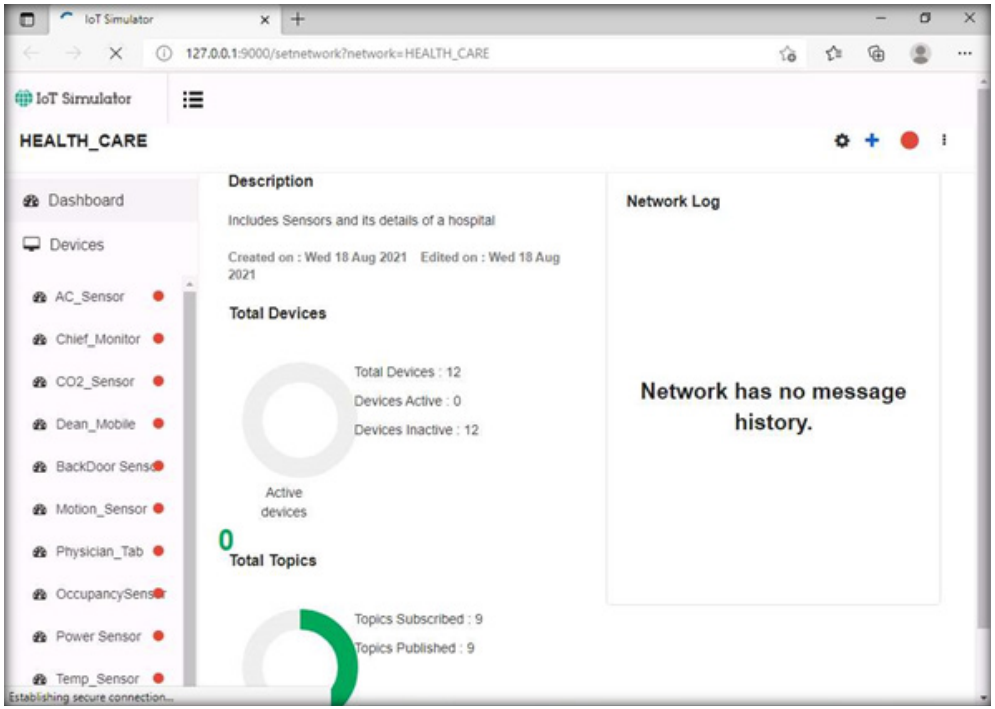
20. Thus, Bevywise IoT Simulator is installed successfully. To launch the **IoT simulator**, navigate to the **C:\Bevywise\IotSimulator\bin** directory and double-click on the **runsimulator.bat** file.



EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

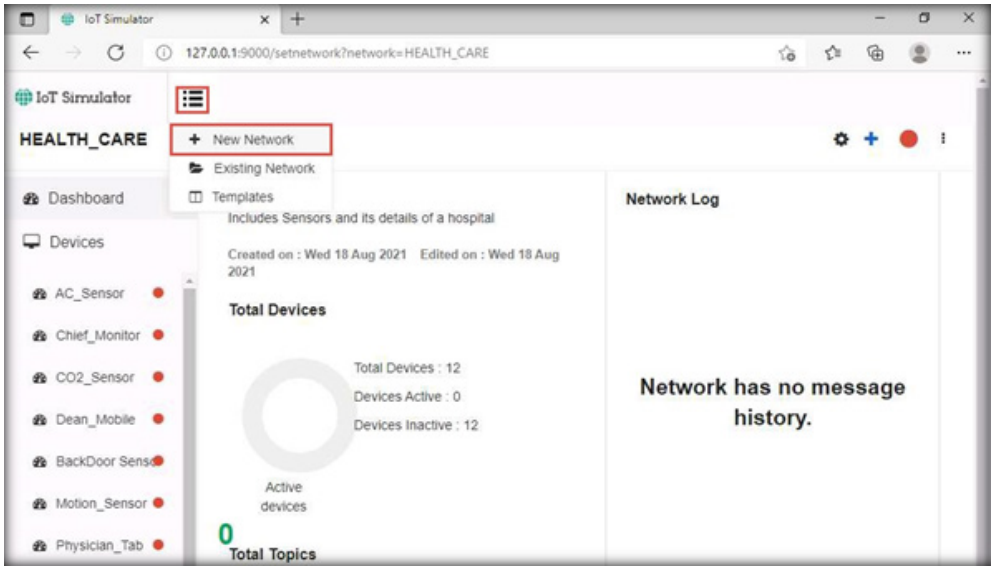
21. Now, the **runsimulator.bat** file opens in the command prompt. If **How do you want to open this?** pop-up appears, select Microsoft Edge browser and click on **OK** to open the following URL: **http://127.0.0.1:9000/setnetwork?network=HEALTH_CARE.**
    Note: If the URL directly opens in Microsoft Edge browser, then continue.
22. The web interface of the IoT Simulator opens in Microsoft Edge browser. In the IoT Simulator, you can view the default network named **HEALTH_CARE** and several devices.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



23. Next, we will create a **virtual IoT network** and **virtual IoT devices**. Click on the **menu** icon and select the +**New Network** option.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

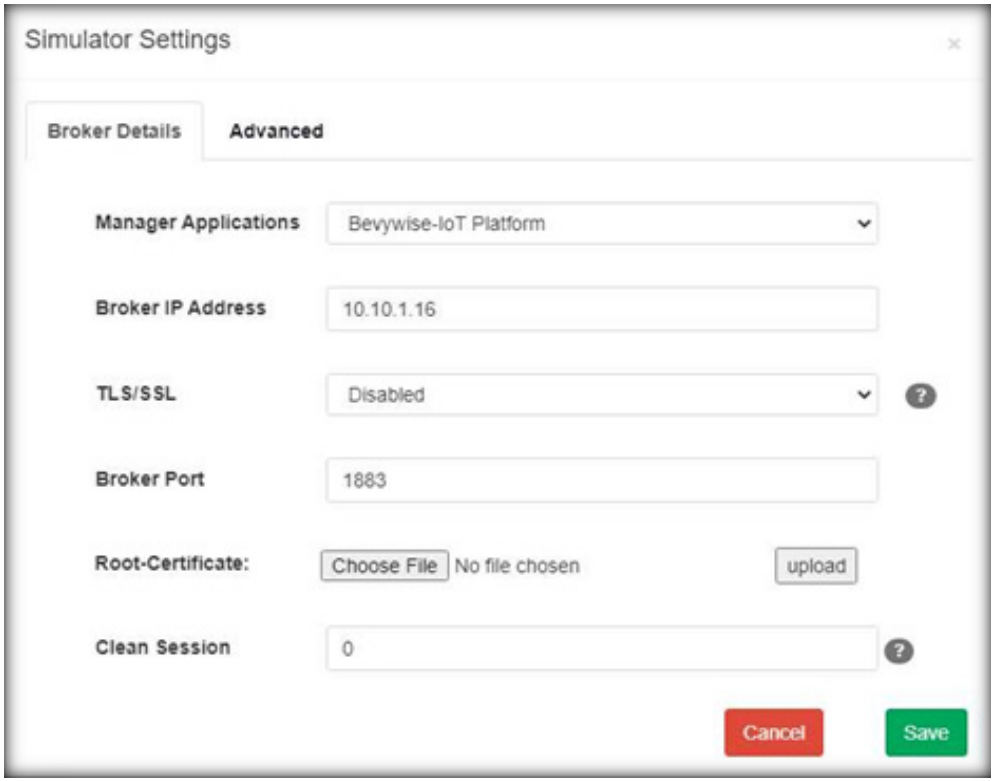


24. The **Create New Network** popup appears. Type any name (here, **CCT_FINANCE_NETWORK**) and description. Click on **Create.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

25. In the next screen, we will setup the **Simulator Settings.** Set the **Broker IP Address as 10.10.1.16** (the IP address of the **Web Server**). Because we have installed the Broker on the web server, the created network will interact with the server using MQTT Broker. Do not change the default settings and click on **Save.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

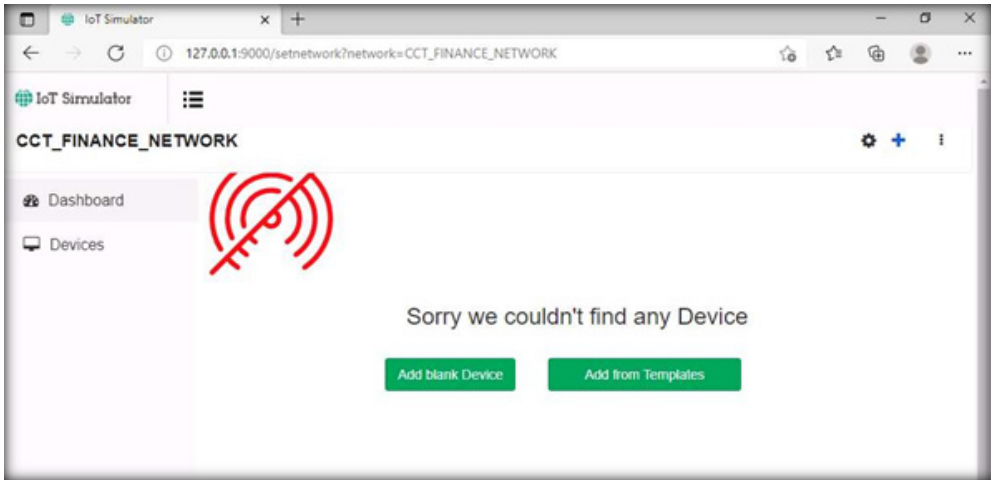


26. To proceed with network creation, click on **Yes.**

Note: If a **Configuration Saved** pop-up appears; click on **OK** to continue. This step completes the creation of a virtual IoT network.

27. To add IoT devices to the created network, click on **Add blank Device.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

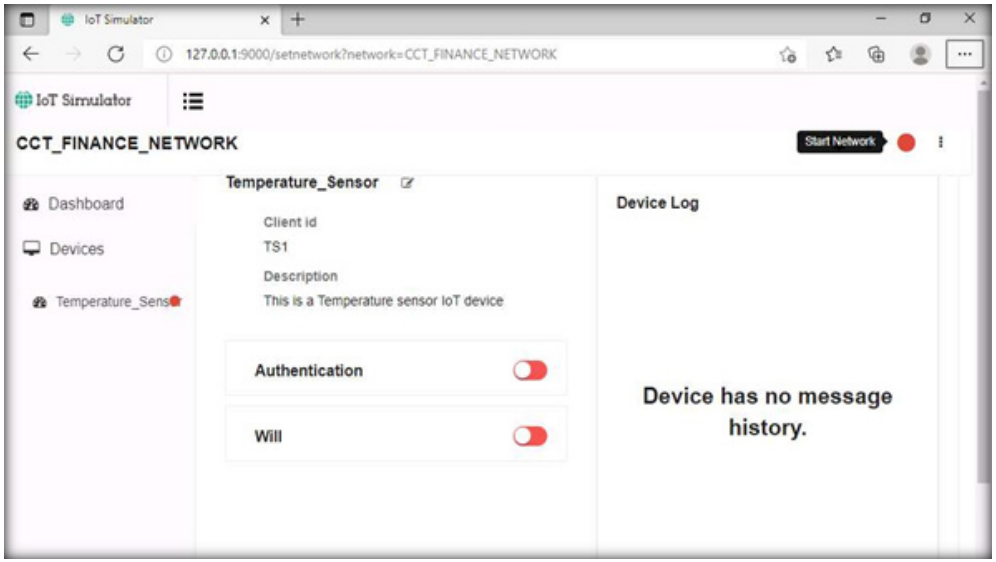


28. The **Create New Device** pop-up appears. Type the device name (here, we used **Temperature_Sensor**), enter Device Id (here, we use **TS1**), provide a **Description** and click on **Save.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

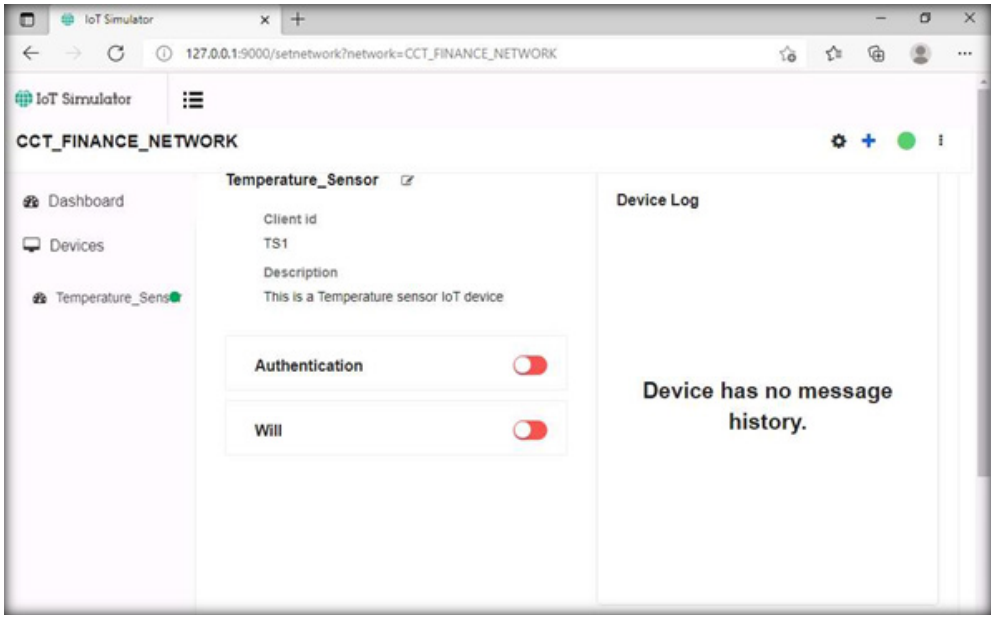
29. The device will be added to the **CCT_FINANCE_NETWORK.**
30. To connect the Network and the added devices to the server or Broker, click on the **Start Network** red color circular icon in right corner.

EXERCISE 1:
SECURE IOT DEVICE
COMMUNICATION
USING TLS/SSL

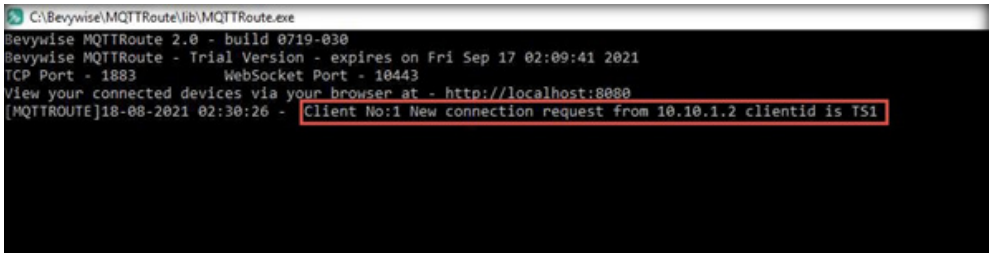


31. When a connection is established between the network and the added devices and the web server or the MQTT Broker, the red button turns into **green.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



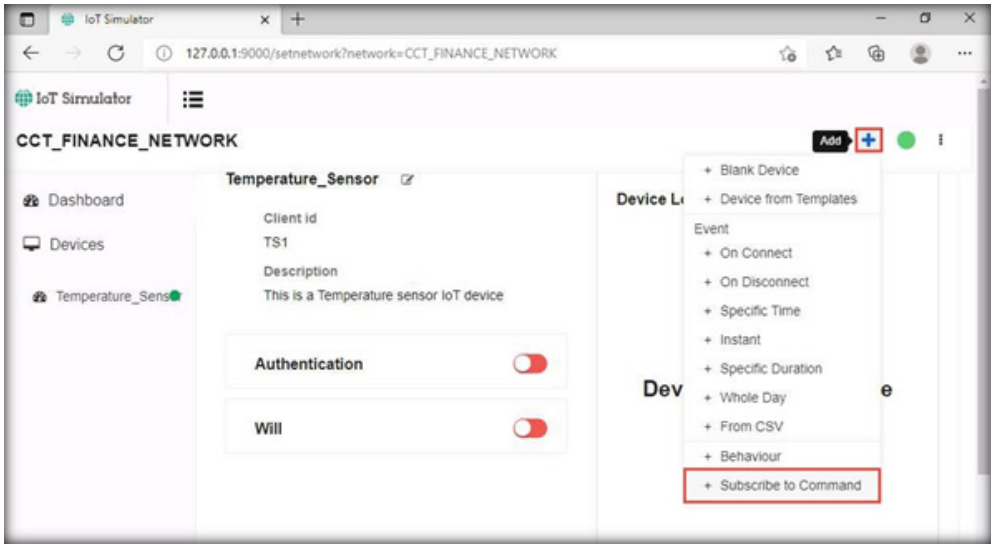
32. Next, switch to the **Web Server** virtual machine. Because the Broker was **left running**, you can see a connection request from machine **10.10.1.2** to device **TS1.**



EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

33. Switch back to the **Admin Machine-1** virtual machine.
34. Next, we will create the Subscribe command for the device **Temperature_Sensor.**
35. Click on the **Plus** icon in the **top right corner** and select the **Subscribe to Command** option.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

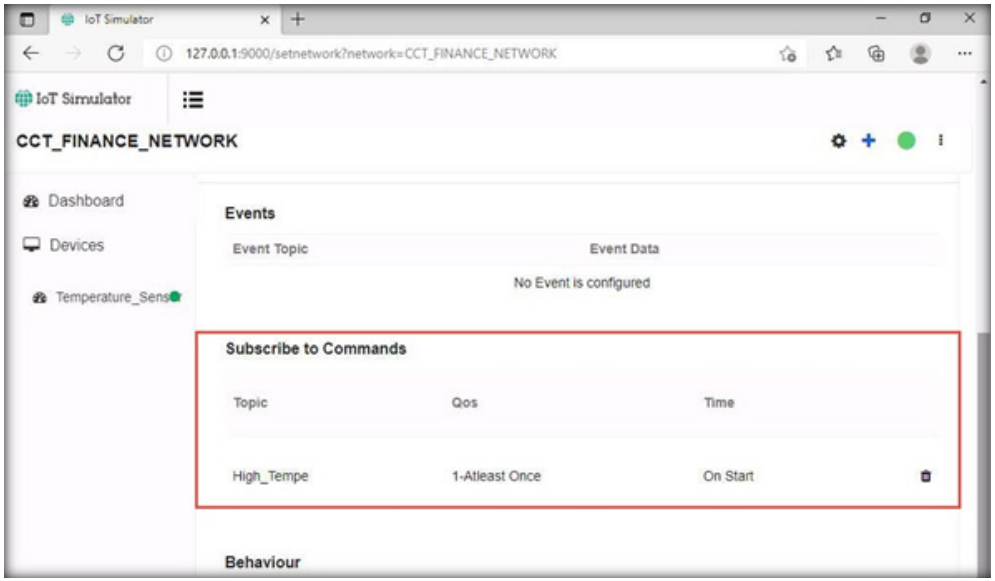


36. The **Subscribe for command - TS1** pop-up appears. Select **On start** under the Subscribe on tab, type High_Tempe under the Topic tab, and select **1 Atleast once** below the **Qos** option. Click on **Save.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

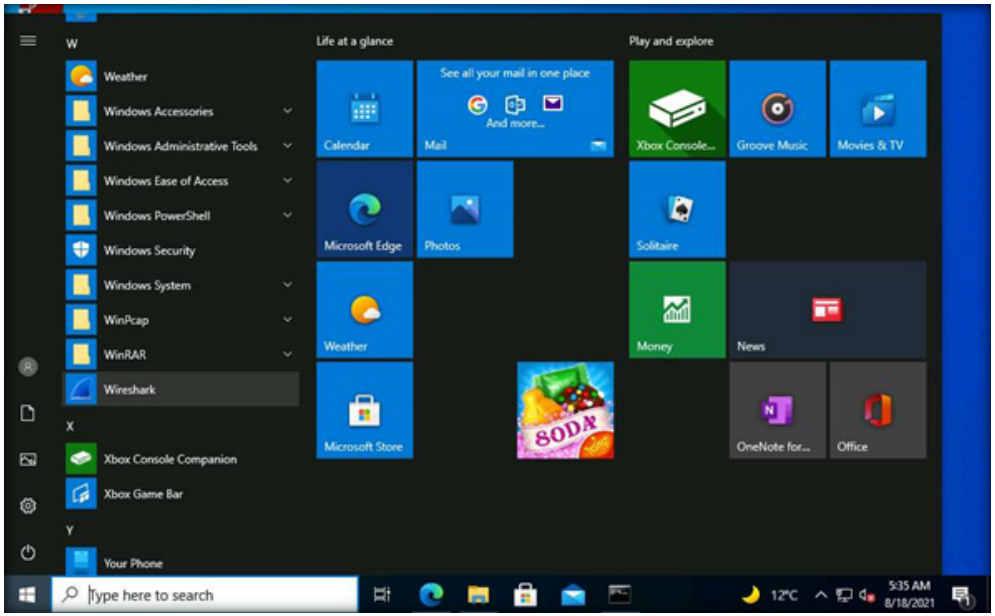
37. Scroll down the page, you can see that the **Topic** has been added under the **Subscribe to Commands** section.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



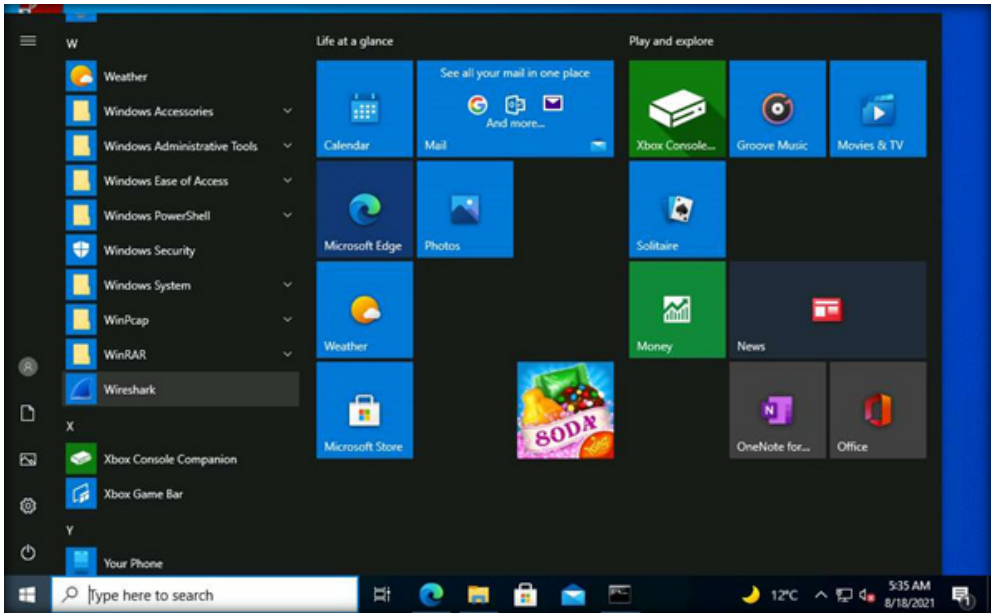
38. Next, we will capture the traffic between the **virtual IoT network and MQTT Broker** to monitor secure communication.
39. Minimize the Edge browser. Click on the Windows **Start** button and launch the **Wireshark** from the application list.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



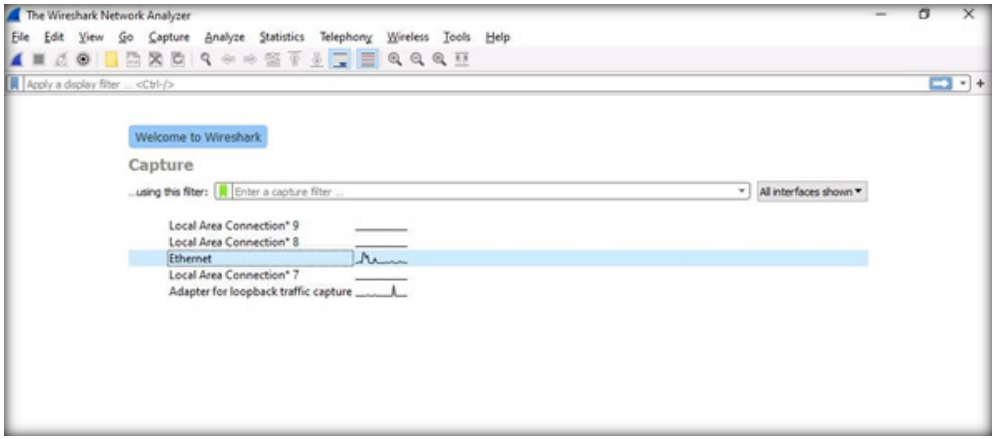
38. Next, we will capture the traffic between the **virtual IoT network and MQTT Broker** to monitor secure communication.
39. Minimize the Edge browser. Click on the Windows **Start** button and launch the **Wireshark** from the application list.

EXERCISE 1:
SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



40. The Wireshark Application window appears, select the **Ethernet** as interface.
    Note: Make sure you have selected interface which has **10.10.1.2** as the IP address.
    Note: If **Software update** popup appears click on **Skip this version.**

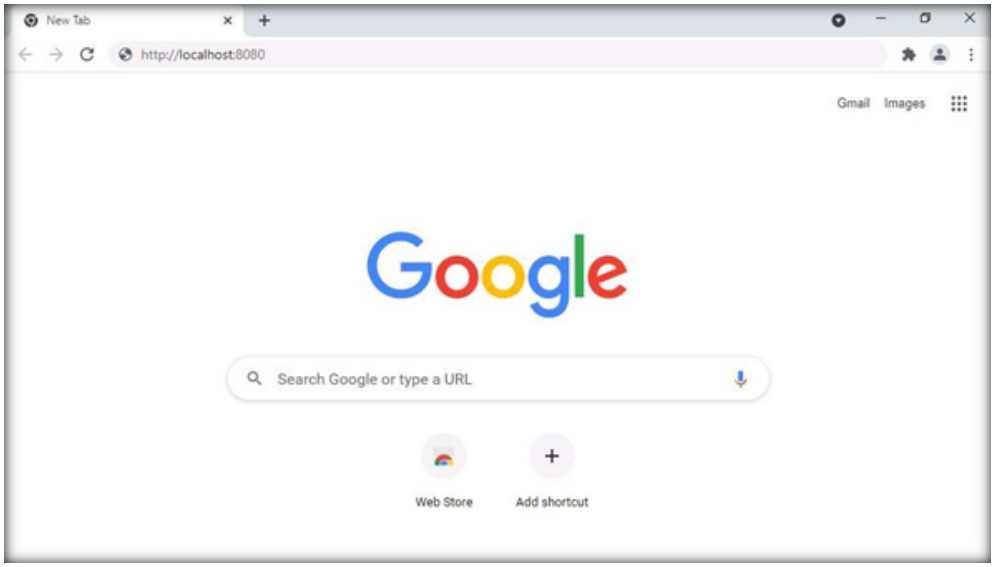
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



41. Click on the **Start capturing packets** icon to start capturing the packets, leave the Wireshark running.
42. Leave the IoT simulator running and switch to the **Web Server** virtual machine.
43. Minimize all opened applications and windows, Open Chrome browser, type **http://localhost:8080** and press **Enter.**

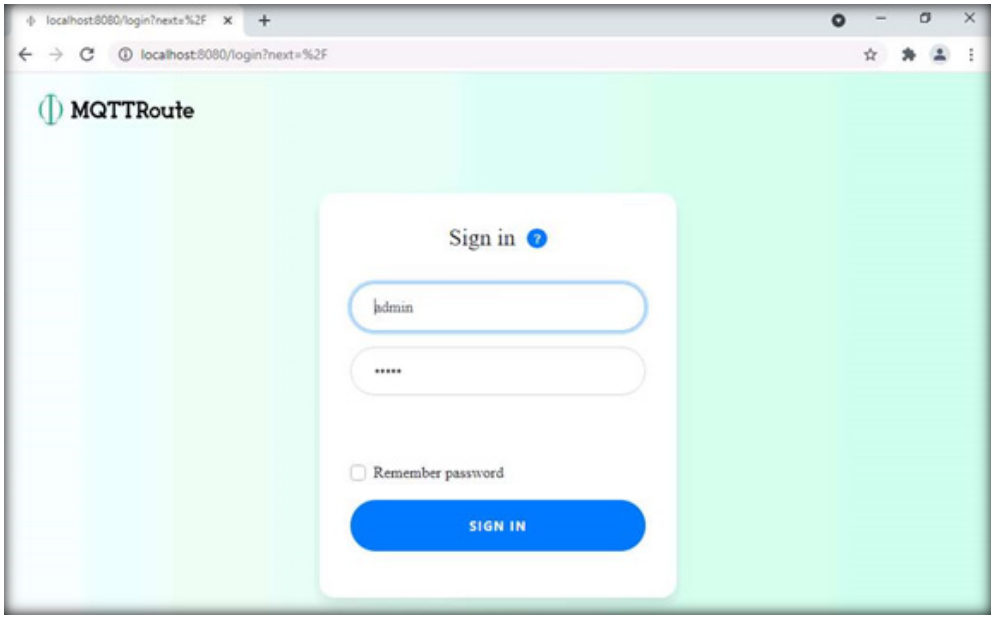
    Note: Do not use Internet Explorer web browser to open the above URL.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



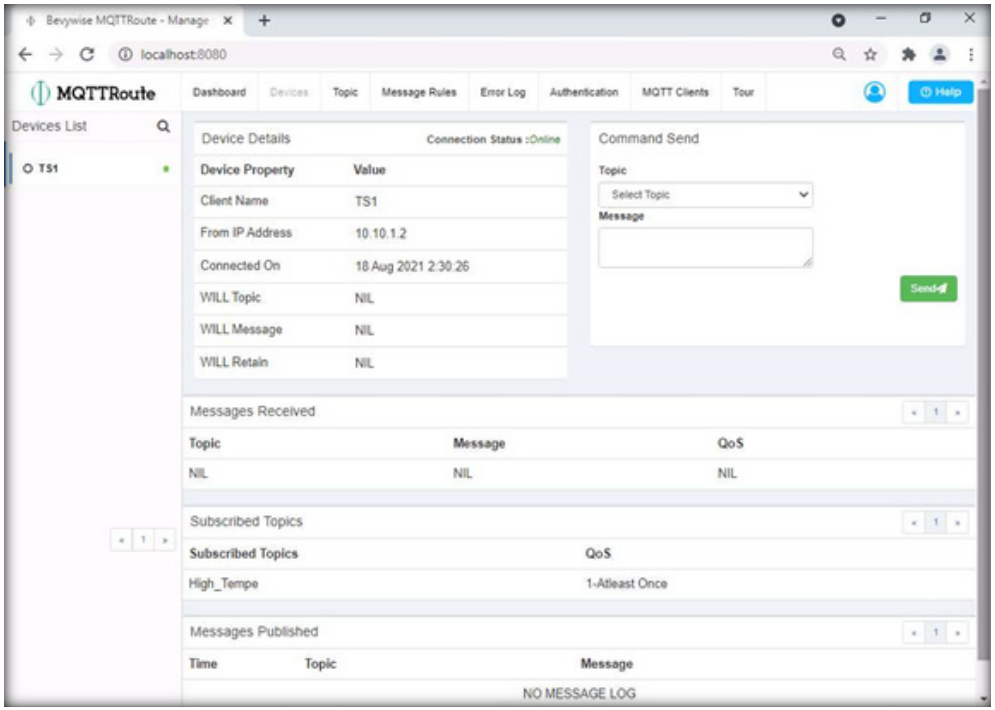
44. As soon as you press **Enter,** the **MQTTRoute Sign in** page appears, leave the default credentials unchanged and click on **SIGN IN.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



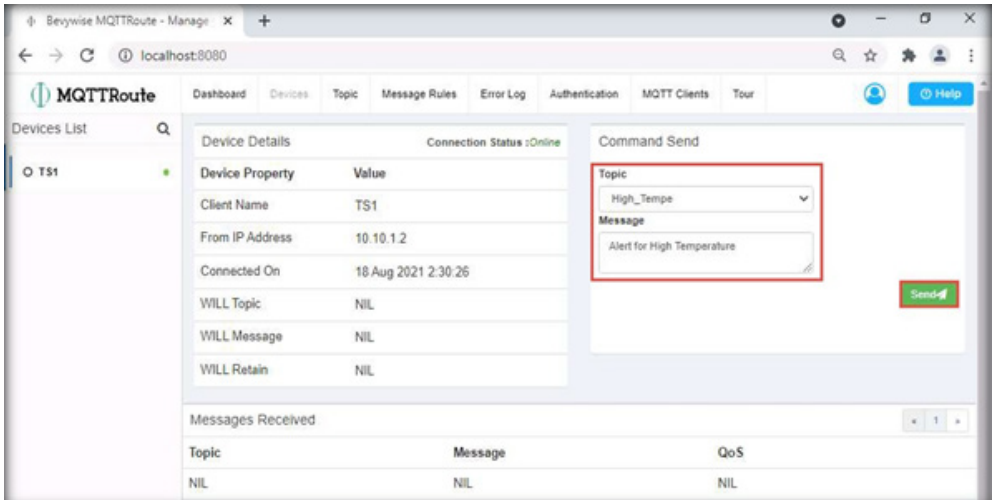
45. Navigate to the **Devices** menu. Now, you can see the connected device **TS1** in the left pane.

EXERCISE 1:
SECURE IOT DEVICE
COMMUNICATION
USING TLS/SSL



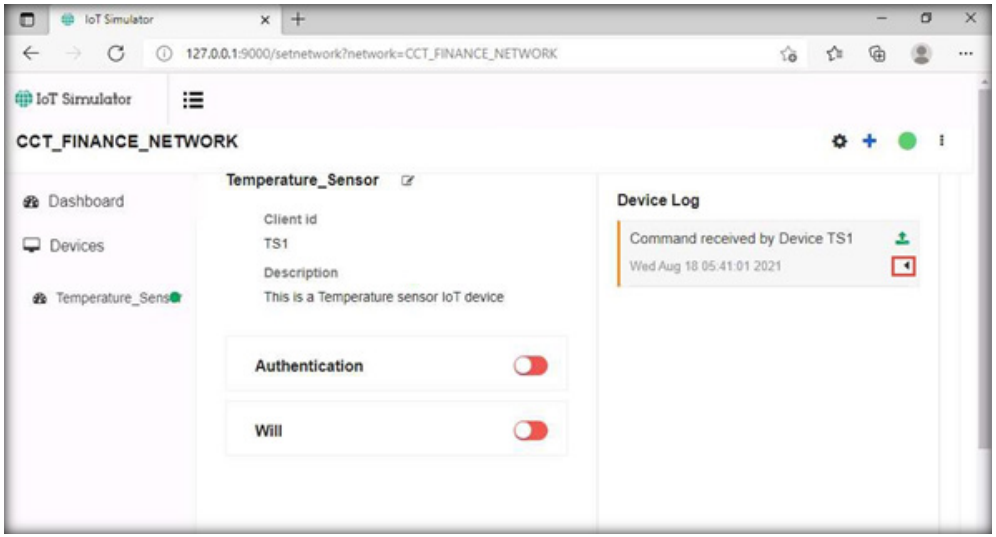
46. Next, we will send the command to **TS1** using the **High_Tempe** topic.
47. Navigate to the **Command Send** section, select **Topic** as **High_Tempe,** type **Alert for High Temperature** and click on the **Send** button.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



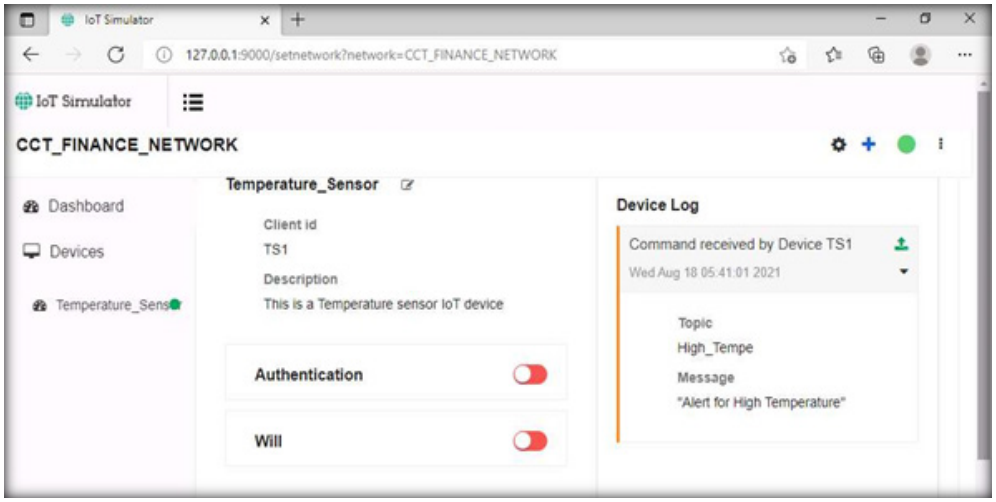
48. An alert pop-up appears, click on **OK.**
49. Thus, the message is sent to the device using this topic.
50. Next, switch to the **Admin Machine-1** virtual machine.
51. We have left the IoT simulator running in the web browser. To see the alert message, maximize the Edge browser and expand the arrow under the connected **Temperature_Sensor**, **Device Log** section.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



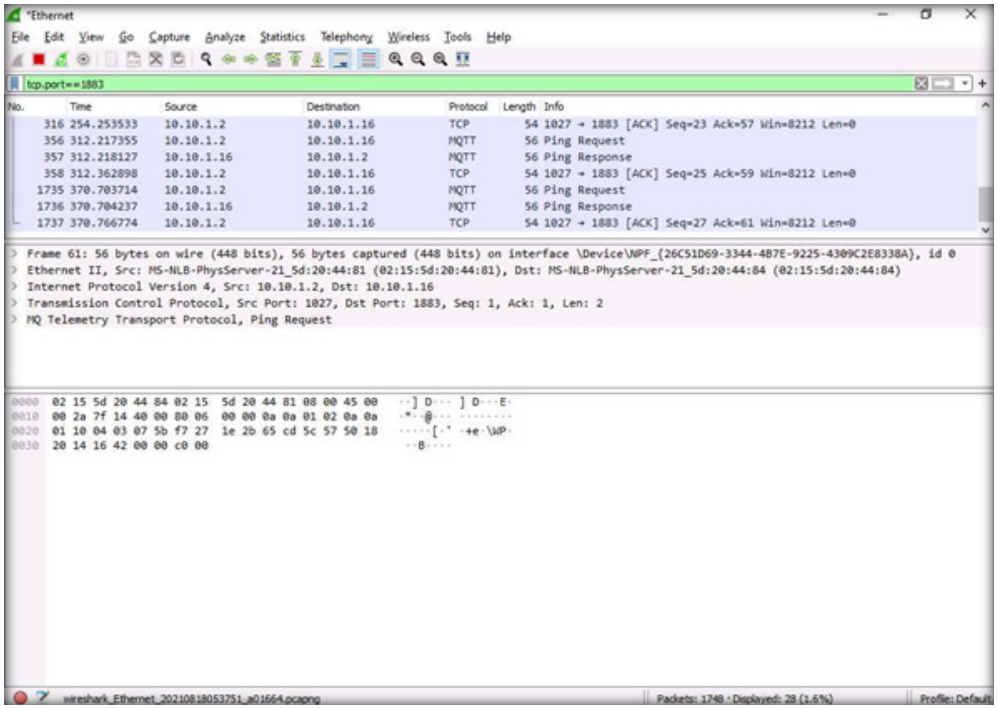
52. You can see the alert message "**Alert for High Temperature".**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



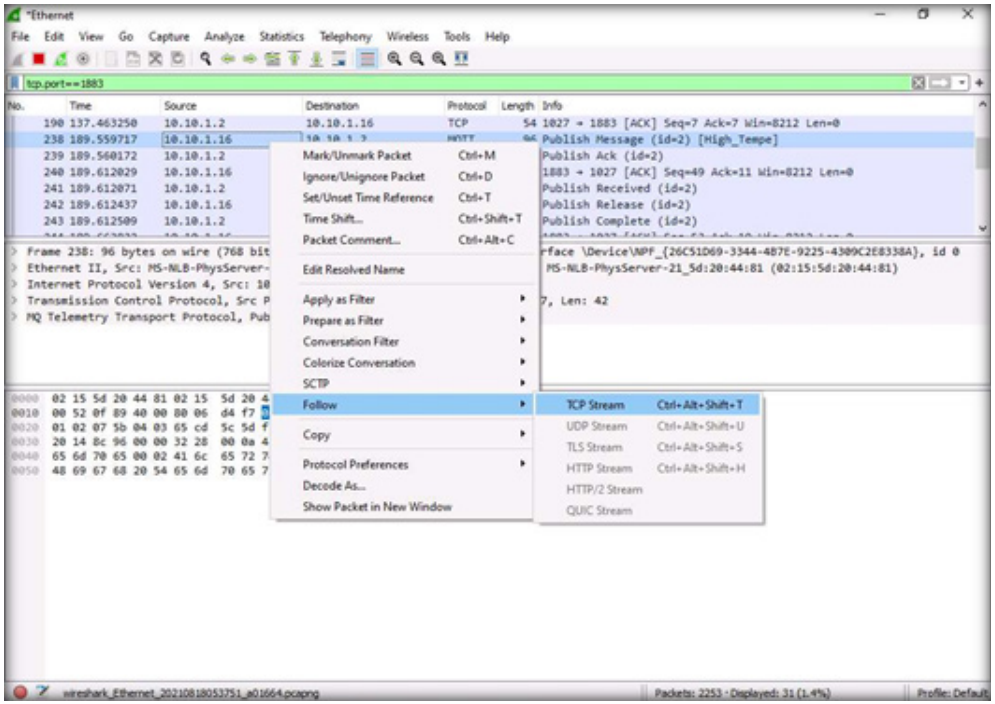
53. To verify the communication, we ran **Wireshark** application. Switch to the Wireshark traffic capturing window.
54. Type **tcp.port==1883** under the **filter** field and press **Enter.** The captured traffic will be filtered.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



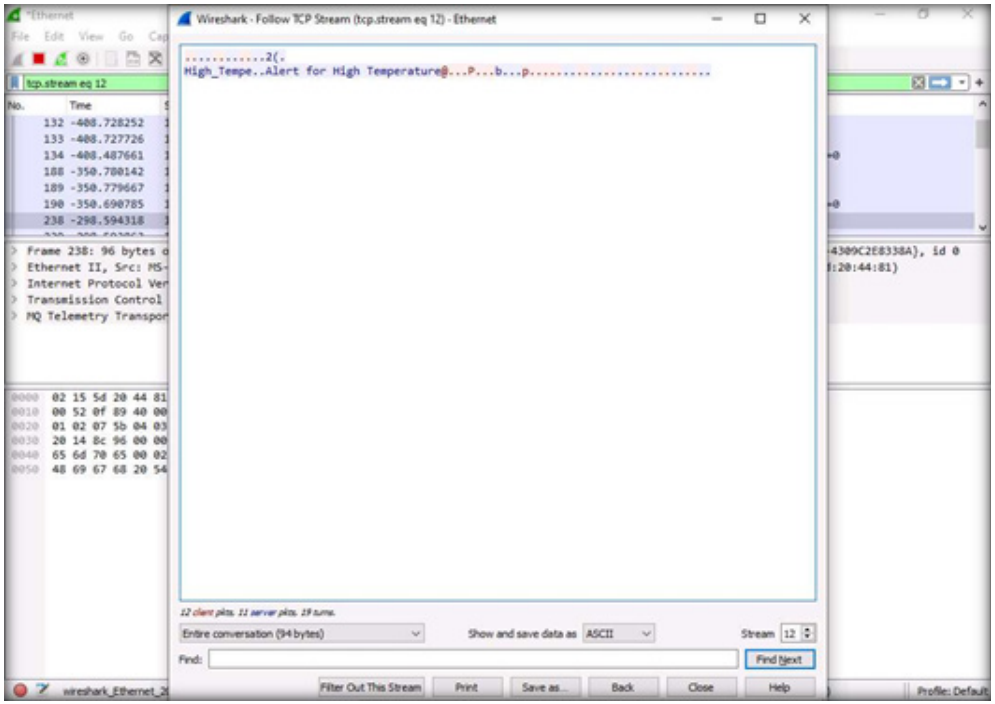
55. Search for **Publish Message**, under the **info** column of Wireshark and **right-click.**
56. From the popup, select **Follow→ TCP Stream.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



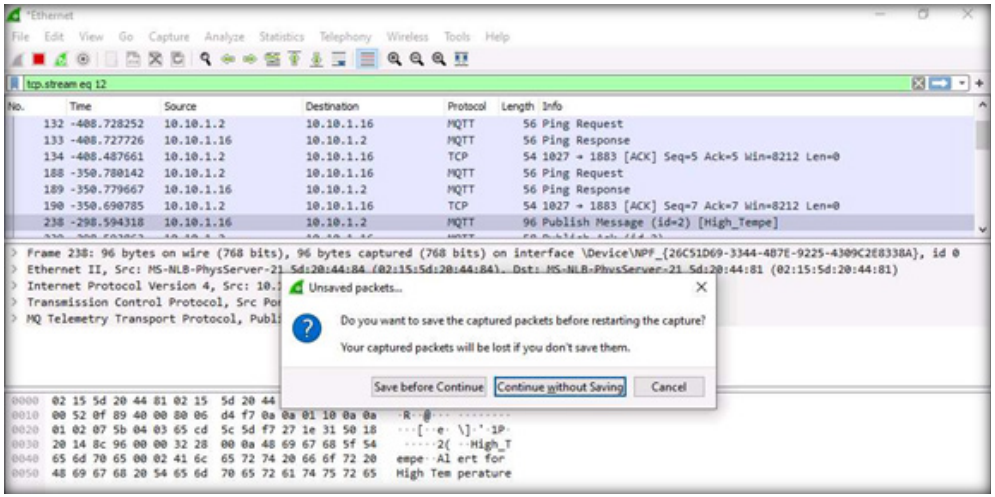
57. You can view the message sent from the server to IoT devices, in **clear plain text**. The attacker can intercept the communication between the server and the device.

EXERCISE 1:
SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



58. Click on **Close** to exit the opened window and click on the green fin icon to **restart** Wireshark. If the **Unsaved packets...** popup appears, click on **Continue without saving.**

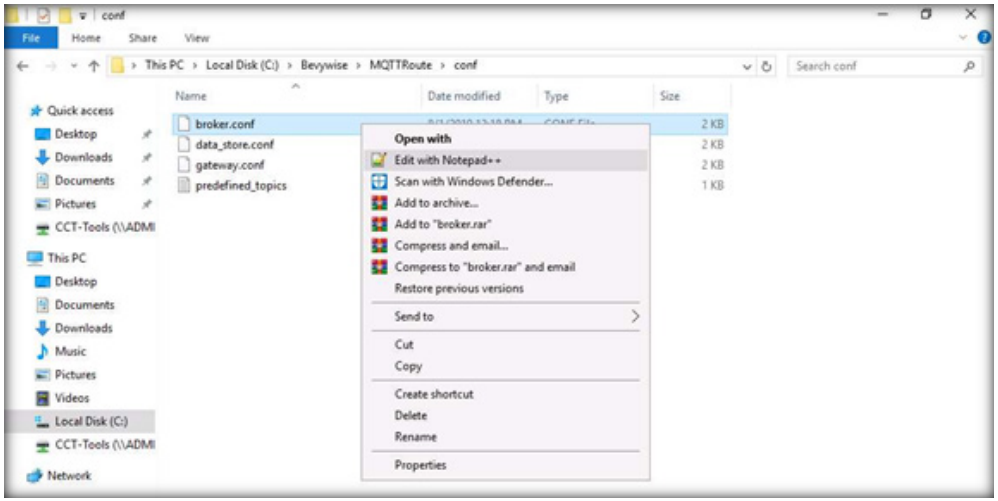
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



59. Next, using self-signed certificates, we will implement **TLS/SSL** on the virtual network to ensure a **secure communica**tion between the device and the server.
60. In this lab, we used the default certificate provided by **Bevywise MQTT Broker.**

    Note: You can use **openssl** to create a **self-signed** certificate.
61. To configure the MQTT Broker for the TLS/SSL communication, switch to the **Web Server** virtual machine.
62. Close the opened **Chrome** Browser, switch to the running **MQTTRoute** Broker in the command prompt and close the command prompt by pressing **CTRL+C** twice.
63. Navigate to the **C:\Bevywise\MQTTRoute\conf** folder and **right-click** on the **broker.conf** file. Click on **Edit with Notepad++.**

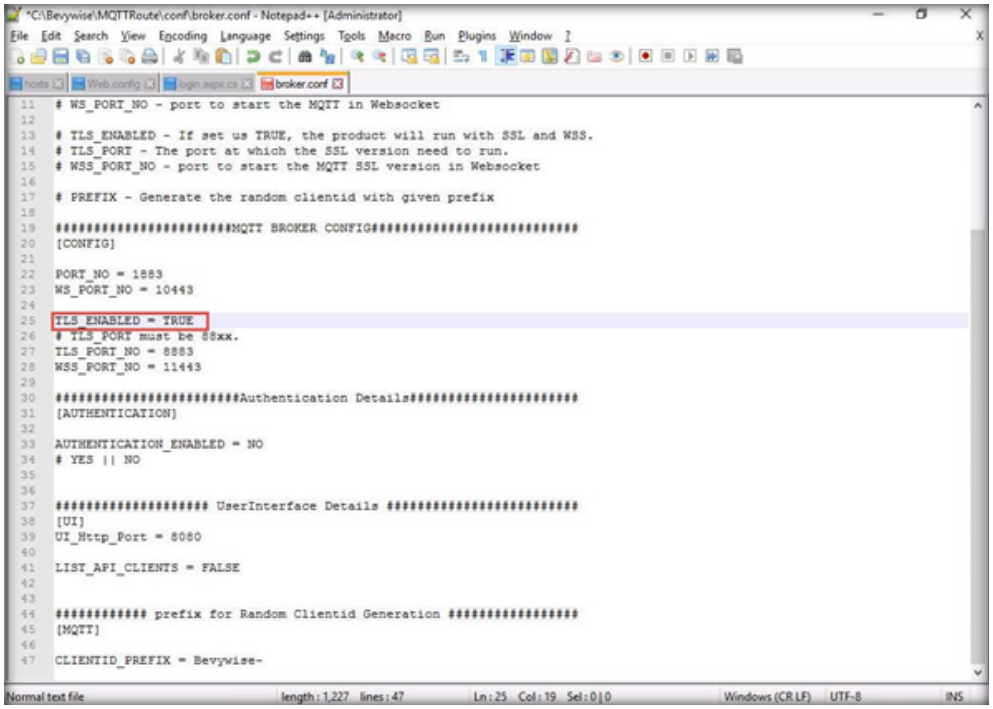
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



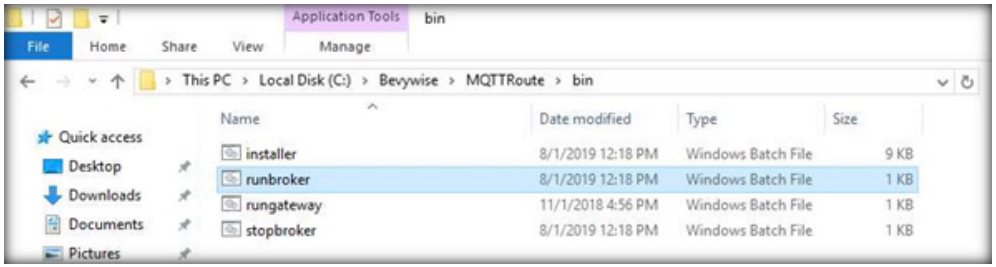
64. The **broker.conf** file opens in Notepad++, go to **line no. 25** and change **TLS_ENABLED=FALSE to TLS_ENABLED=TRUE**; subsequently click on Save and close the file.

Note: If a **Notepad++ update** pop-up appears, click **No.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

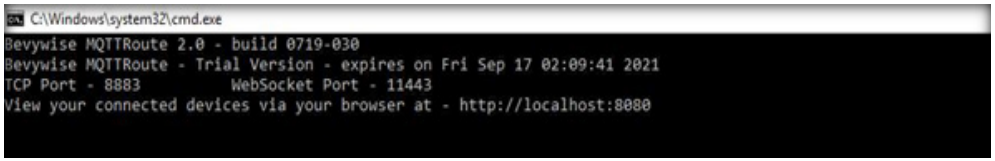


65. Navigate to **C:\Bevywise\MQTTRoute\bin** and double-click on the **runbroker.bat** file.



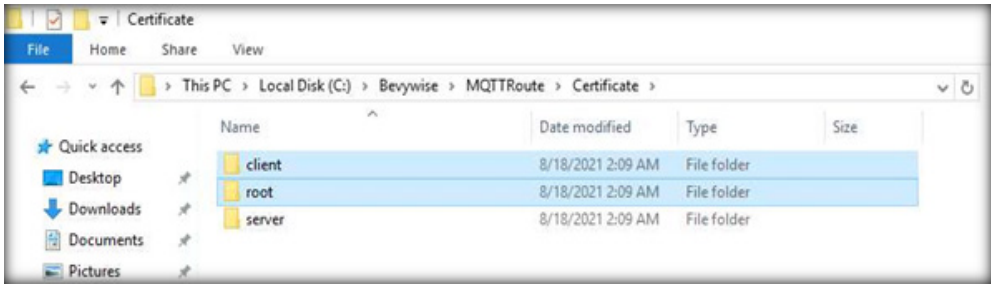
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

66. Upon the execution of the Bevywise MQTTRoute Broker, it can be observed that the TCP port use port **8883** for communicating with IoT virtual network devices over **TLS/SSL** communication.



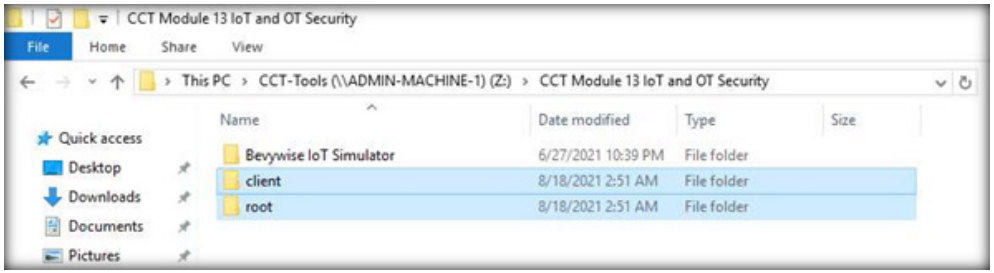
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

67. Leave the Bevywise MQTTRoute Broker running. To copy the certificates from the **server** to the **client,** navigate to **C:\Bevywise\MQTTRoute\Certificate** and copy the **Client** and **root** folders.



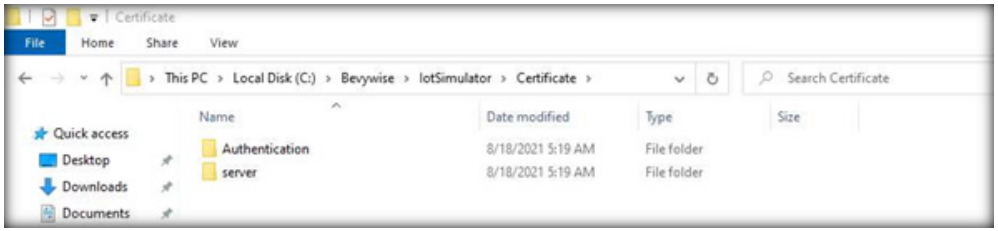
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

68. Navigate to **Z:\CCT Module 13 IoT and OT Security** and press **Enter.** Paste the **copied client**, root folders.



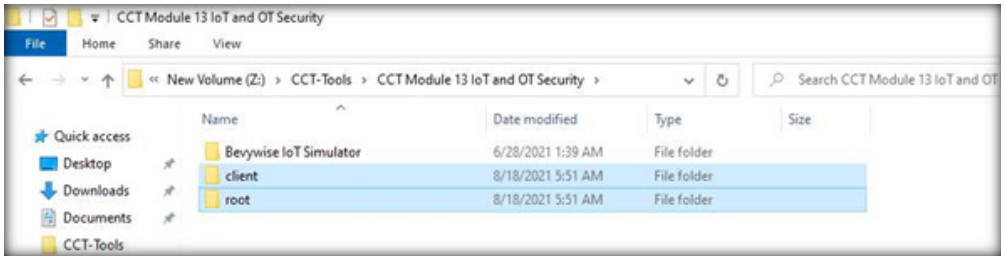
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

69. Switch to the **Admin Machine-1** virtual machine.
70. Minimize the running Wireshark. Navigate to the **C:\Bevywise\IotSimulator\Certificate** and delete the existing **client** and **root** folders.



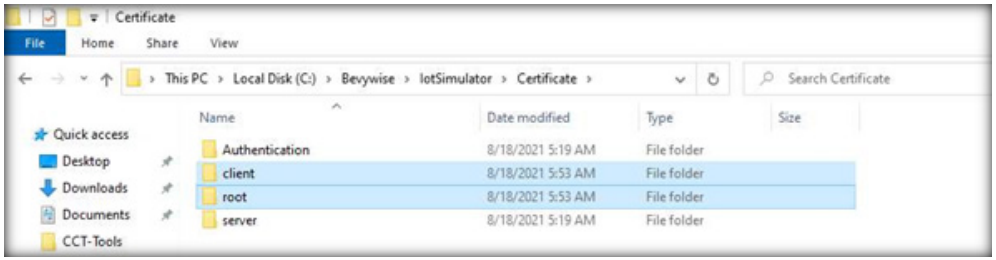
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

71. Open **File Explorer** and navigate to **Z:\CCT-Tools\CCT Module 13 IoT and OT Security** and copy the **client** and **root** folders.



EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

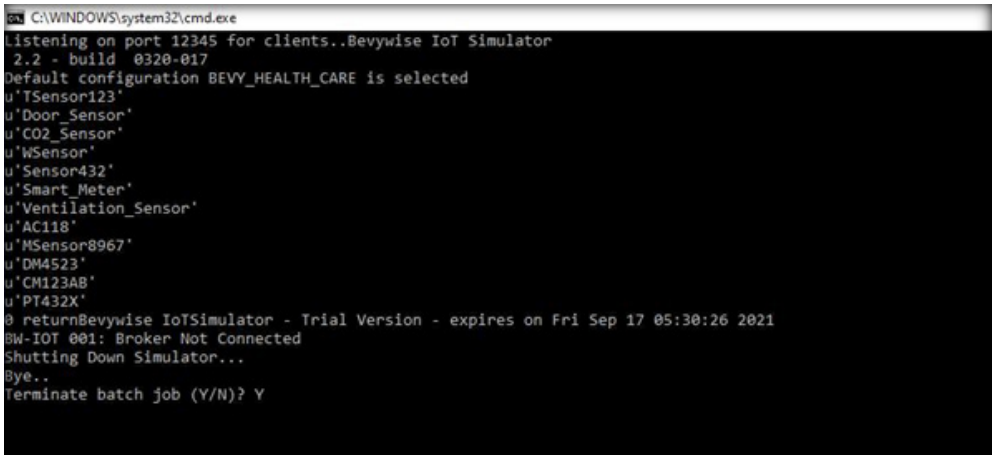
72. Again, navigate to **C:\Bevywise\IotSimulator\Certificate** and **paste** the copied **client** and **root** folders here.



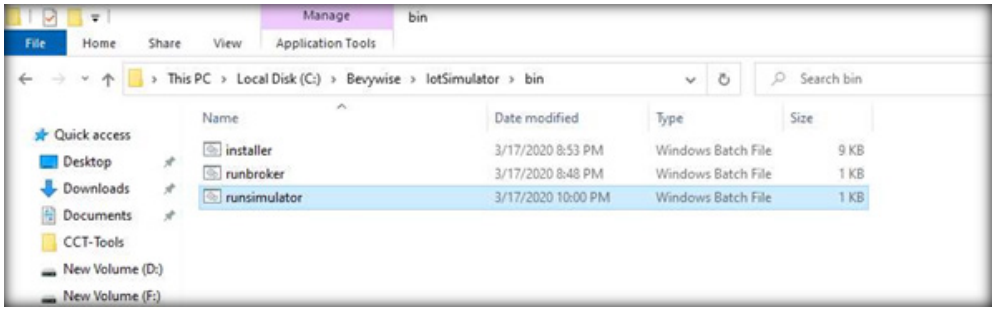
EXERCISE 1:
SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

73. Thus, we have shared the certificate and root key from server to the client machine (**Web Server to Admin Machine-1**) for secure communication of the IoT simulator.
74. To connect to the virtual IoT network and change the network configuration to TLS\SSL, switch to the running **IoT simulator** in command prompt and press **CTRL+C** twice.
75. This will generate a prompt for shutting down the simulator. Type **Y** and press **Enter** to close the command prompt. Close the browser running the IoT simulator.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



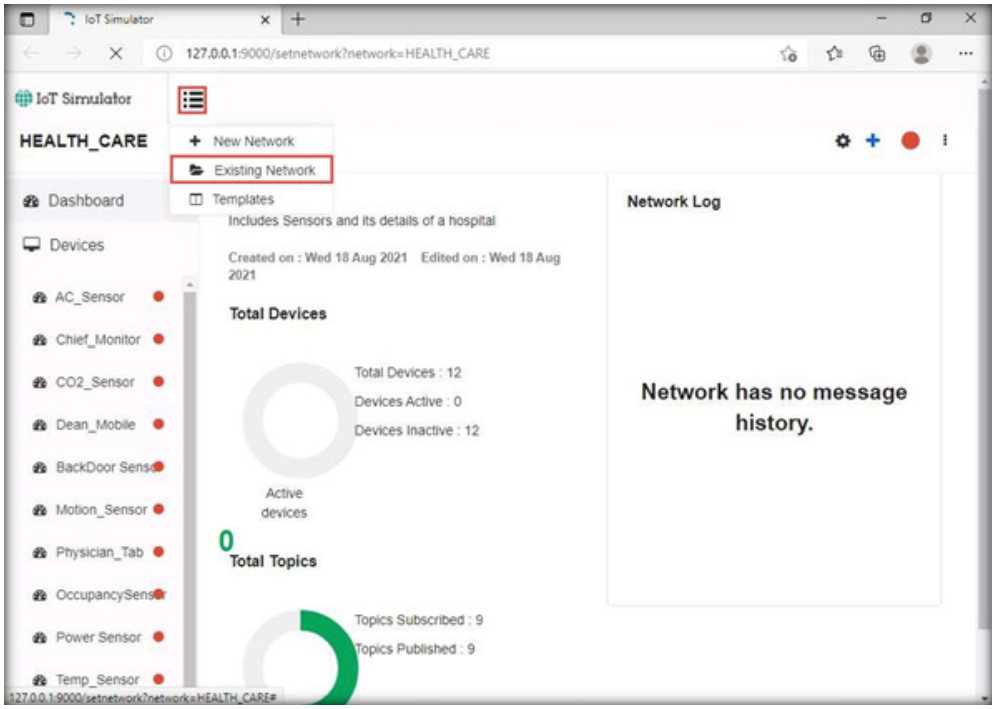
76. Now we will **re-run** the IoT simulator. Navigate to the **C:\Bevywise\IotSimulator\bin** and double-click on the **runsimulator.bat** file.



EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

77. Now, the IoT simulator will be opened in the default web browser.
78. The default network is connected. To switch to **CCT_FINANCE_NETWORK,** click on the **menu** icon and select **Existing Network.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



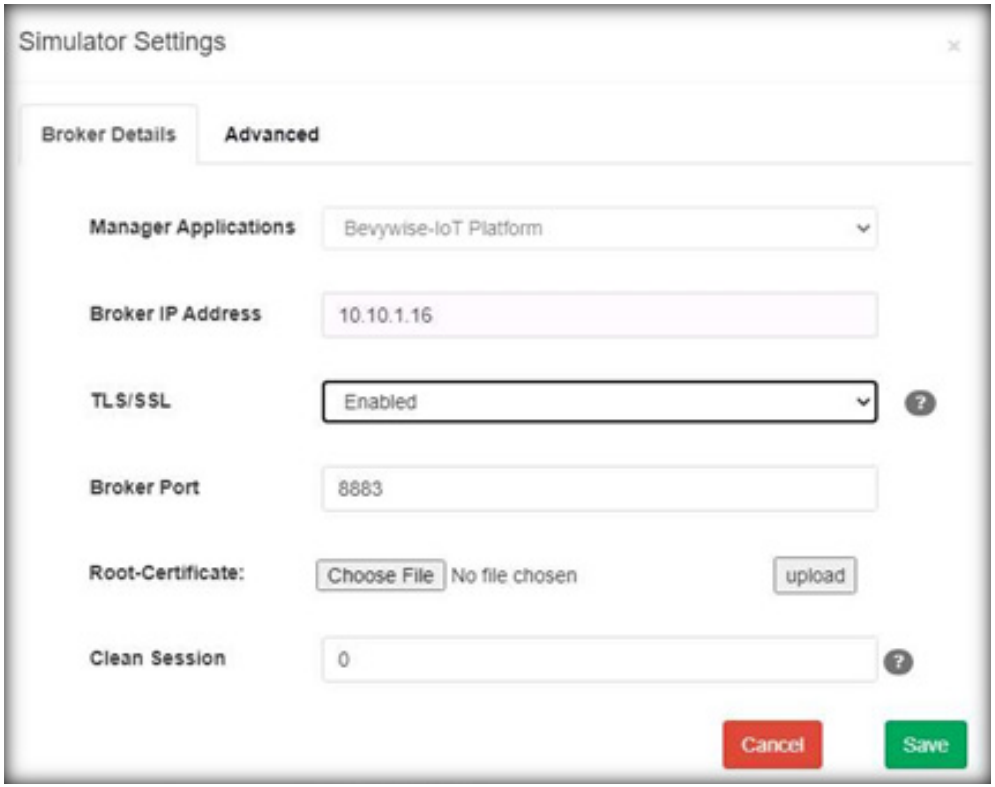
79. A **Choose Network** popup appears. Under **Choose Network**, select **CCT_FINANCE_NETWORK** and click on **Open.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

80. Click on the **setting** icon to change the **network setting.**

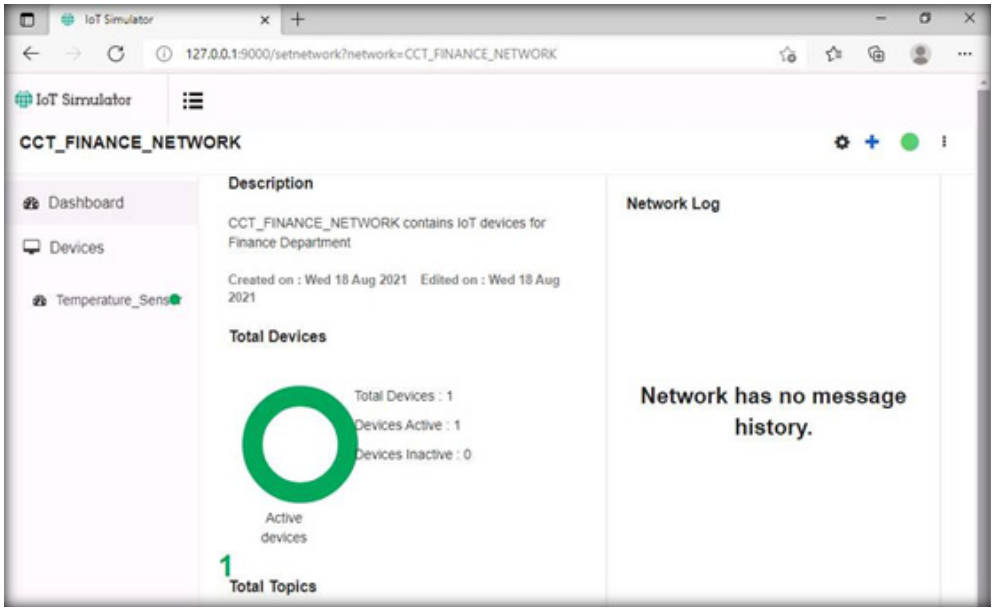
81. The **Simulator Settings** window appears. Click on **Enabled for TLS/SSL**. This will automatically change the Broker Port to **8883.** Click on **Save.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



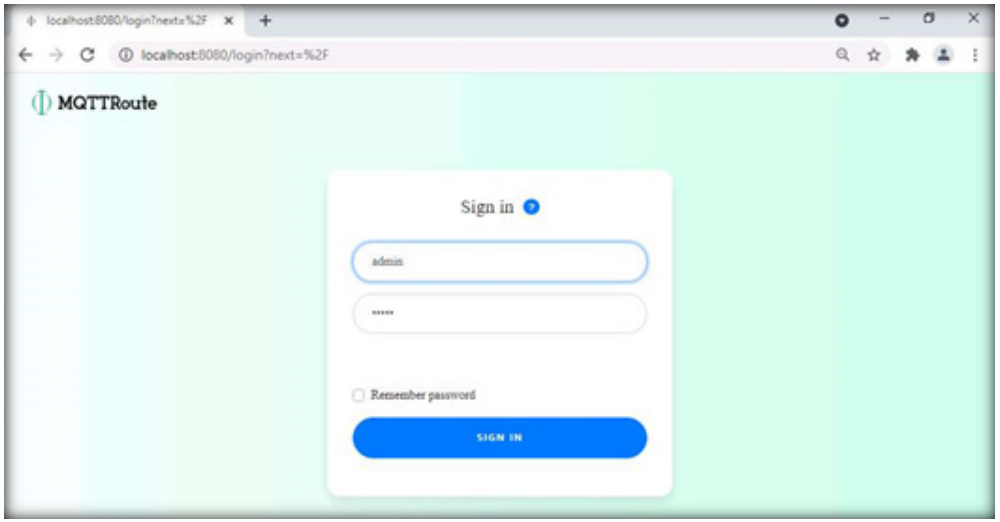
82. The **Configuration Saved** popup appears. Click on **OK.**
83. Next, we will start the network. Click on the **red button** in the top right corner.
84. The red button turns into green, indicating both the network connection and the device connection.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



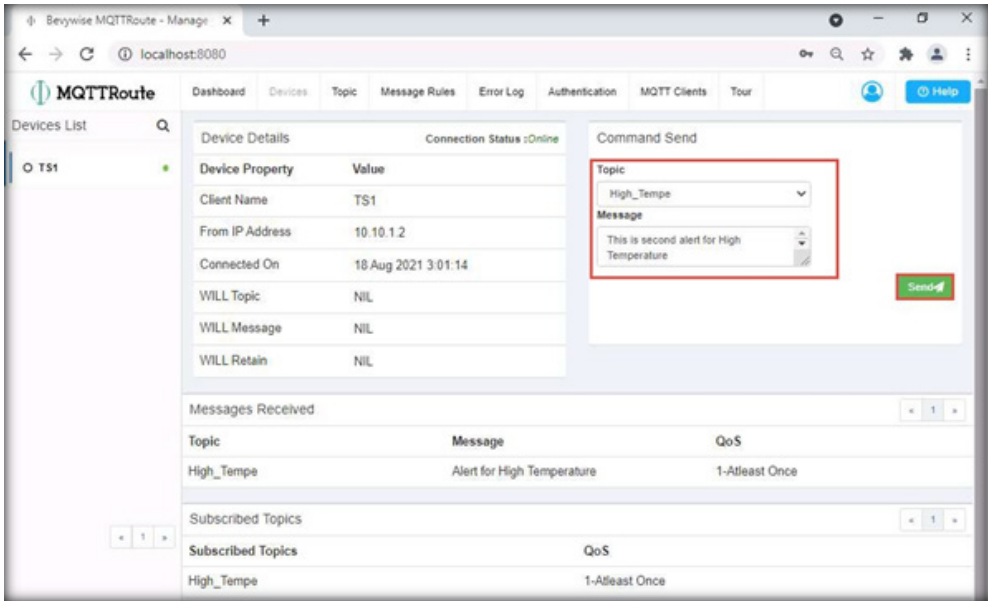
85. Leave the IoT simulator running.
86. Switch to the **Web Server** virtual machine.
87. Open the **Chrome** browser and type **http://localhost:8080** and press **Enter.** This will redirect you to the MQTTRoute sign-in page. Do not change the default credentials and click on **SIGN IN.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



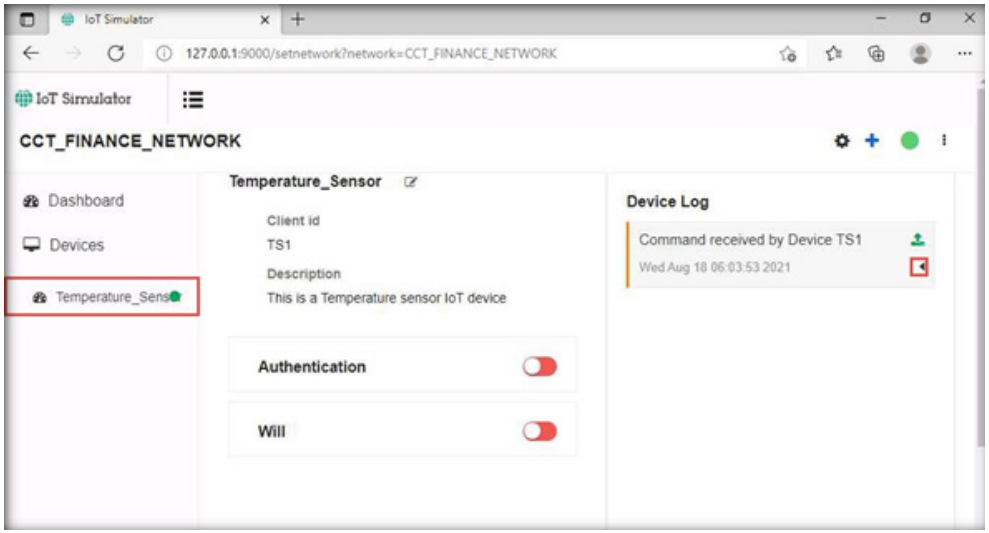
88. Select the **Devices** menu to see the connected device **TS1.**
89. Next, we will send the same command to **TS1** using the **High_Tempe** topic.
90. Go to the **Command Send** section, select **High_Tempe** under the **Topic** tab, type "**This is second alert for High Temperature**" in the message tab and click on **Send.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



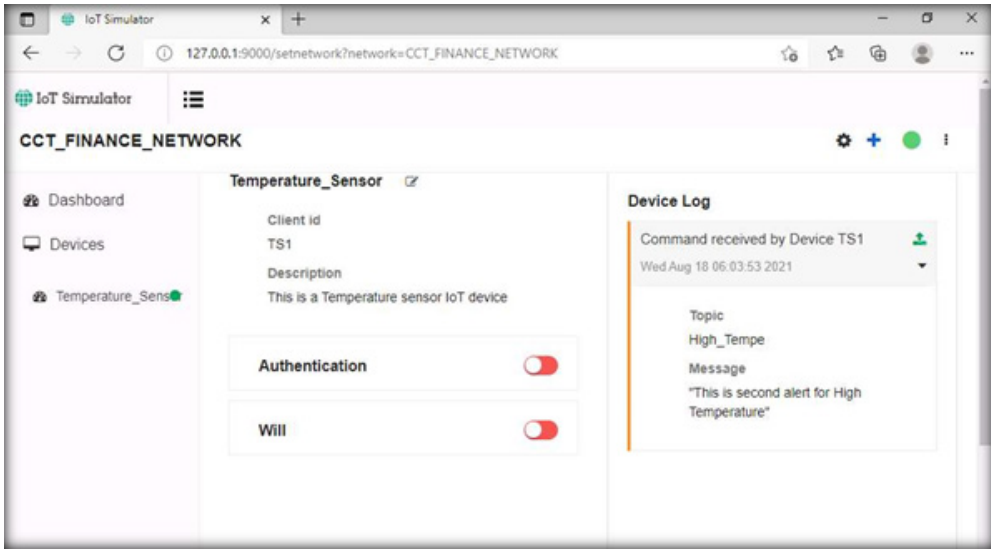
91. The alert popup appears; click on **OK.** The message is sent to the device using this topic.
92. Next, switch to the **Admin Machine-1** virtual machine.
93. We have left the IoT simulator running. To see the alert message, select the **Temperature_Sensor** IoT device and expand the **arrow** under the **Device Log** section.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



94. You can see the alert message that we sent from the Web Server "**This is second alert for High Temperature**".

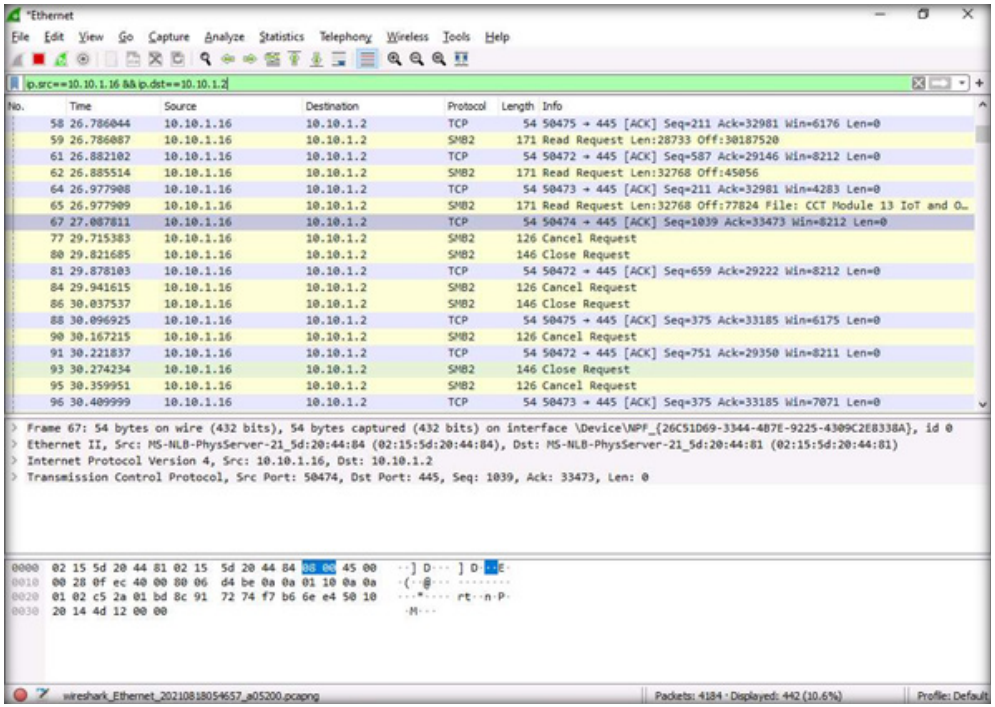
EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



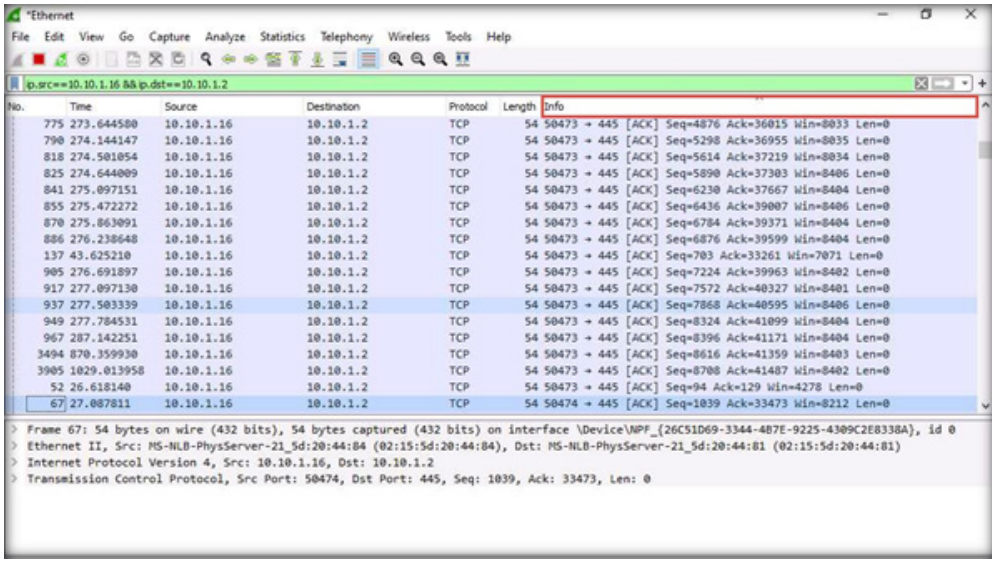
95. We have left the Wireshark running to verify the communication. Switch to running **Wireshark** application, click stop the traffic capturing.

96 To filter the traffic, type **ip.src==10.10.1.16 && ip.dst==10.10.1.2** in the filter field and press **Enter.**

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



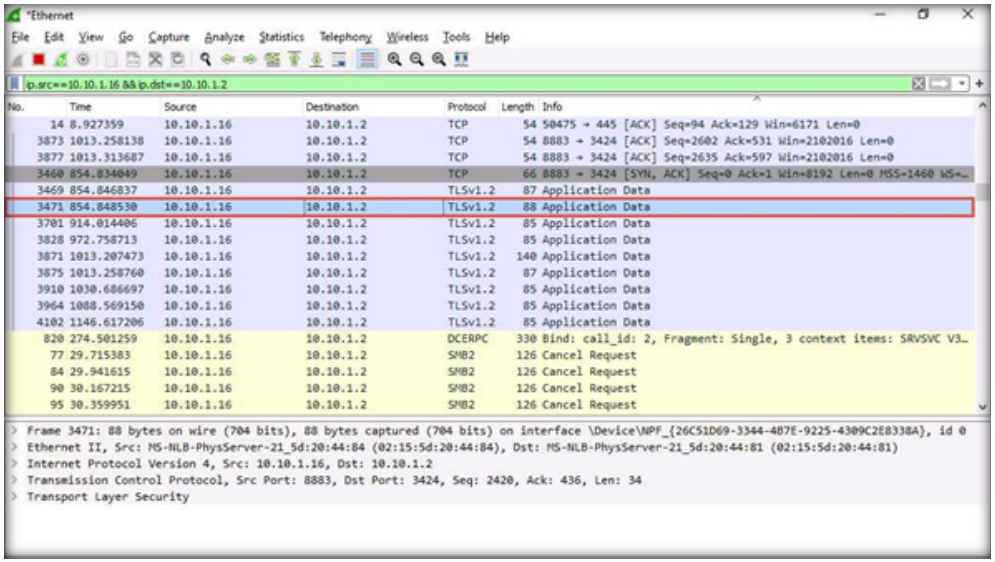
97. Click on the **Info** tab in the Wireshark filter table.



EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

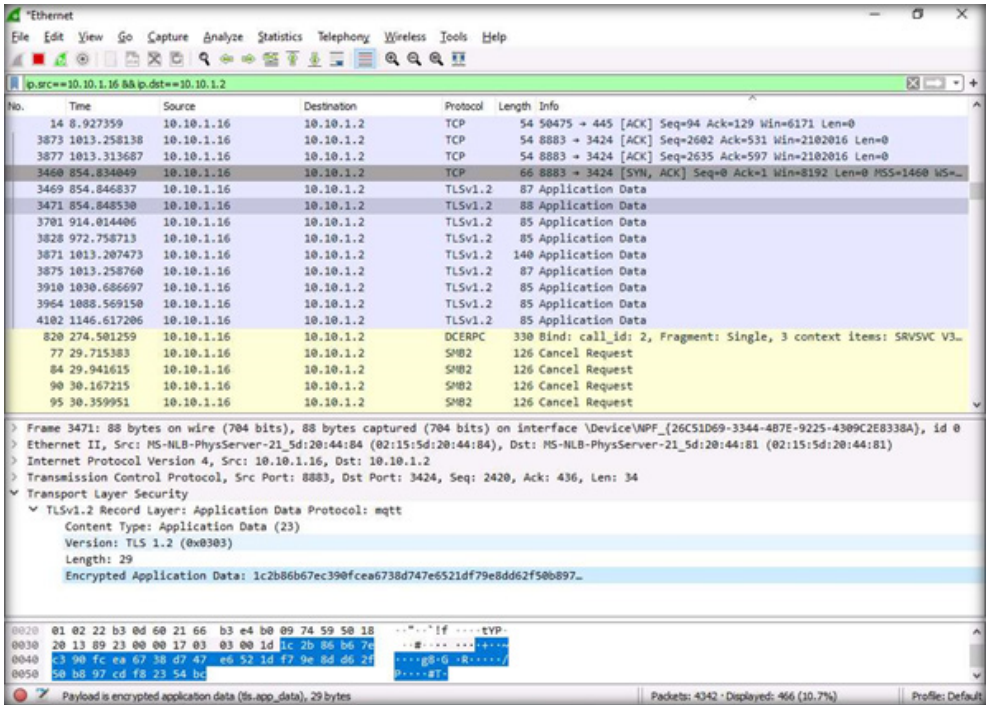
98. The packets will be sorted; now, select any **TLSv1.2** as the protocol and **Application Data** as the info packet.

EXERCISE 1:
SECURE IOT DEVICE COMMUNICATION USING TLS/SSL

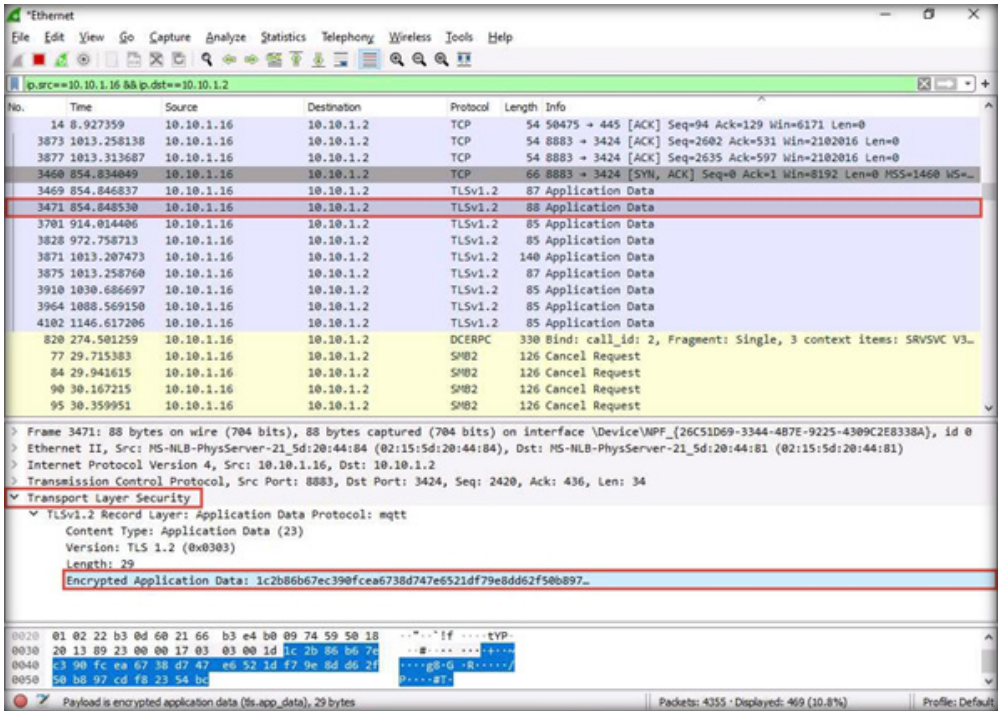


99. You can find the details of the **Encrypted Application data** under the **Transport Layer.**

EXERCISE 1:
SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL



100. By implementing the aforementioned steps, a security professional can securely configure IoT devices and protect them from malware infections within the network.
101. This concludes the demonstration showing how to secure IoT device communication using TLS/SSL.
102. Close all open windows.
103. Turn off **Admin Machine-1, Web Server** and **PfSense Firewall** virtual machines.

EXERCISE 1: SECURE IOT DEVICE COMMUNICATION USING TLS/SSL


EC-Council