





## CERTIFIED CYBERSECURITY TECHNICIAN







Chapter 13: IoT and OT Security

Exercise 1: Secure IoT Device Communication using TLS/SSL







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:



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.













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.



				Application Tools	Bevywise IoT Simulator			-
File	Home	Share	View	Manage				
$\leftarrow \   \rightarrow$	· ↑	« CCT	-Tools (\\A	DMIN-MACHINE-1)	(Z:) > CCT Module 13 IoT and OT Secu	urity > Bevywise lo	T Simulator	~ Č
			Name	^	Date modified	Туре	Size	
🖈 Qu	ick access		Bevy	wise_loTSimulator_W	/in_64 5/17/2020 8:08 PM	Application	34,417 KB	
J D	ownloads	*	🔕 Bevy	wise_MQTTRoute_W	in_64 7/6/2020 1:12 AM	Application	29,509 KB	
🔁 D	ocuments	*						
F Pi	ictures	*						
<b>=</b> 0	CT-Tools (	\ADMI						





- 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.









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



C:\Bevywise\MQTTRoute\lib\MQTTRoute.exe

Bevywise MQTTRoute 2.0 - build 0719-030 Bevywise MQTTRoute - Trial Version - expires on Fri Sep 17 02:09:41 2021 TCP Port - 1883 WebSocket Port - 10443 View your connected devices via your browser at - http://localhost:8080





- 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.
- 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.









- 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.



Setup - IoTSimulator 2.1	- 0 ×
	Completing the IoTSimulator Setup has finished installing IoTSimulator on your computer. The application may be launched by selecting the installed shortcuts. Click Finish to exit Setup.
	Einish





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.



🖸 🔤 🖛	_		Manage	bin
File Home	Share	View	Application Tools	
← → * ↑ 📕	> This	PC > Local	Disk (C:) > Bevywi	ise > lotSimulator > bin → ♂ , ○ Search bin
		Name	^	Date modified Type Size
A Quick access		installer	r	3/17/2020 8:53 PM Windows Batch File 9 KB
Desktop	#	runbrok	cer	3/17/2020 8:48 PM Windows Batch File 1 KB
Downloads	1	💿 runsimu	ulator	3/17/2020 10:00 PM Windows Batch File 1 KB
Documents	#			
CCT-Tools				
- New Volume (	(D:)			
- New Volume	(F:)			
Videos				





- 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.









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.



Γ.	
CCT_FINANCE_NETWORK	
CCT_FINANCE_NETWORK contains IoT devices for Finance Department	
	11
	CCT_FINANCE_NETWORK CCT_FINANCE_NETWORK contains IoT devices for Finance Department





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**.



Simulator Settings			×
Broker Details	Advanced		
Manager Applie	cations	Bevywise-IoT Platform 🗸	
Broker IP Addr	ess	10.10.1.16	
TLS/SSL		Disabled	0
Broker Port		1883	
Root-Certificate	e:	Choose File No file chosen upload	
Clean Session		0	0
		Cancel	Save







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.









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**.



eate New Device		
Device Name	Temperature_Sensor	
Device Id	TS1	
Description	This is a Temperature sensor IoT device	
	Cancel	Sav





- 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.









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**.



S C:\Bevywise\MQTTRoute\ib\MQTTRoute.exe
Bevywise MQTTRoute 2.0 - build 0719-030
Bevywise MQTTRoute - Trial Version - expires on Fri Sep 17 02:09:41 2021
TCP Port - 1883 WebSocket Port - 18443
View your connected devices via your browser at - http://localhost:8080
[MQTTROUTE]18-08-2021 02:30:26 - Client No:1 New connection request from 10.10.1.2 clientid is TS1





- 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.









36. The **Subscribe for command - TSI** 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**.



source for command - Te	51	
Subscribe on		
On start 👻		
Topic	Qas	
High_Tempe	1 Atleast once	





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



← → C ① 12							
IoT Simulator	I .						
CT_FINANCE_NETW	ORK			0	+	•	1
8 Dashboard	Events						
Devices	Event Topic	Even	t Data				
֎ Temperature_Sens●		No Event is config	ured				
	Subscribe to Comma	nds					
	Торіс	Qos	Time				
	High_Tempe	1-Atleast Once	On Start				





- 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 JSING 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 JSING 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.



Ele Edit View Go Capture Analyze Statistics Telephony Wireless Jools Help		
Acoly a display filter <col-></col->	-	•
Welcome to Wireshark		
Capture		
Local Area Connection* 9 Local Area Connection* 8		
Ethernet		
Local Area Connection* 7		
Адарсет гогороваск transc capture		





- 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.









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



· → C () localhost8080/login?n	ext=%2F	* * *
() MQTTRoute		
	Sign in 👩	
	ledmin	
	Remember password	





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

EXERCISE 1: Secure Iot device communication Using tls/ssl

() MQTTR	oute	Dashboard	Devices.	Topic	Message Rules	Error Log	Authent	tication	MQTT Clients	Tour		٢	œ	Help
Devices List	Q	Device D	etails		Connec	tion Status :0	sine	Com	mand Send					
O TS1		Device Property Value					Topic							
		Client Name			TS1			Sel	lect Topic		~			
		From IP A	ddress	10	10.1.2			Messa	age .					
	Connected On 18 Aug 2021 2:30:26													
	WILL Topic		NI									Sen	4	
	WILL Mes	sage	NI											
	WILL Reta	in	NI	í.										
		Messages	Received										×	1 -
		Topic			M	essage				QoS				
		NIL			N	L				NIL				
a 1 a		Subscribed	Topics											1 >
	1 3	Subscribed Topics						QoS						
		High_Tempe						1-Atleas	st Once					
		Messages	Published											1 .
		Time	Top	Nic			N	Message	e					





- 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

Bevywise MQT	TRoute - Ma	anage X	+									0	-	0		×
← → C	<li>Iocalho</li>	ost8080										Q	\$	*		:
() MQTTR	oute	Dashboard	Devices	Торіс	Message Rules	Error Log	Authen	tication	MOTT Clients	Tour			8	C	Help	
Devices List	Q	Device D	Details		Connec	tion Status :On	ine	Com	mand Send							
O TS1		Device Pr	roperty	Va	lue			Topic			_					
		Client Nar	me	TS	51			Hig	h_Tempe		~					
		From IP A	ddress	10	10.1.2			Aler	ige I for High Tempera	ture						
		Connecte	d On	18	Aug 2021 2:30:2	6					h			_		
		WILL Topi	ic	N	L									Sen	4	
		WILL Mes	ssage	N	Ĺ											
		WILL Ret	ain	N	L											
		Messages	Received												1.0	
		Topic			м	essage				QoS						
		NIL			N	L				NIL						







- 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.

















- 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.



Source 10.10.1.2 10.10.1.2 10.10.1.16 10.10.1.2 10.10.1.2 10.10.1.2 10.10.1.2 10.10.1.2	Destination 10.10.1.16 10.10.1.16 10.10.1.2 10.10.1.16 10.10.1.16	Protocol L TCP MQTT MQTT TCP	ength Info 54 1027 + 1883 [ACK] Seq-23 Ack=57 Win=8212 Len=0 56 Ping Request 56 Ping Response	
10.10.1.2 10.10.1.2 10.10.1.16 10.10.1.2 10.10.1.2 10.10.1.2 10.10.1.16 10.10.1.2	10.10.1.16 10.10.1.16 10.10.1.2 10.10.1.16 10.10.1.16	TCP MQTT MQTT TCP	54 1027 → 1883 [ACK] Seq=23 Ack=57 Win=8212 Len=0 56 Ping Request 56 Ping Response	
10.10.1.2 10.10.1.16 10.10.1.2 10.10.1.2 10.10.1.16 10.10.1.16	10.10.1.16 10.10.1.2 10.10.1.16 10.10.1.16	MQTT MQTT TCP	56 Ping Request 56 Ping Response	
10.10.1.16 10.10.1.2 10.10.1.2 10.10.1.16 10.10.1.2	10.10.1.2 10.10.1.16 10.10.1.16	TCP	56 Ping Response	
10.10.1.2 10.10.1.2 10.10.1.16 10.10.1.2	10.10.1.16	TCP	F4 1007 - 1003 F1CV1 FOF 1F0 100010 10	
10.10.1.2 10.10.1.16 10.10.1.2	10.10.1.16		54 1027 * 1885 [ALK] Seq=25 ACK=59 WIN=8212 Len=0	
10.10.1.16		MQTT	56 Ping Request	
10.10.1.2	10.10.1.2	MQTT	56 Ping Response	
	10.10.1.16	TCP	54 1027 + 1883 [ACK] Seq=27 Ack=61 Win=8212 Len=0	
4 84 92 15 5d 20 44 0 90 80 96 00 90 0a 7 5b 77 27 1e 2b 65 0 90 c0 00	81 88 60 45 60 ··] Ba 81 82 8a 8a ·· cd 5c 57 50 18 ···	D ] DE -@ -[-* -+e-\WP B		
	HS-NLB-PhysServer-21 Version 4, Src: 10.1 rrol Protocol, Src Por isport Protocol, Pring 4 84 82 15 5d 20 44 8 86 86 66 80 80 81 8 80 66 80 86 8 80 66 80 80 8 80 66 80	MS-NLB-PhysServer-21_5d:20:44:81 (02:15: Version 4, 5rc: 10.10.1.2, Dst: 10.10.1. rol Protocol, 5rc Port: 1027, Dst Port: 1 isport Protocol, Ping Request 4 84 02 15 5d 20 44 81 08 00 45 00 0 00 80 06 00 00 8a 00 102 0a 0a 0 00 c0 00	MS-NLB-PhysServer-21_5d:20:44:81 (02:15:5d:20:44:81), Version 4, Src: 10.10.1.2, Dst: 10.10.1.16 rol Protocol, Src Port: 1027, Dst Port: 1883, Seq: 1, isport Protocol, Ping Request 4 84 02 15 5d 20 44 81 08 00 45 00 ···] D···] D···E 0 00 80 06 00 00 8 00 10 20 8 0 ···] D···] D···E 0 00 c0 00 ···] 125 5d 5c 5c 5c 5c 5c 5s 18 0 00 c0 00 ···] ····	MS-NLB-PhysServer-21_5d:20:44:81 (02:15:5d:20:44:81), Dst: NS-NLB-PhysServer-21_5d:20:44:84 (02:15:5d:20:44:84) Version 4, Src: 10.10.1.2, Dst: 10.10.1.16 rol Protocol, Src Port: 1027, Dst Port: 1883, Seq: 1, Ack: 1, Len: 2 isport Protocol, Ping Request 4 84 02 15 5d 20 44 81 08 00 45 00 ···] D···E· 0 00 80 00 00 0 8 a 01 02 0 a 0 ···] D···E· 0 00 80 00 00 0 8 a 01 02 0 a 0 ···] D···E· 0 00 c0 00 ···B····





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



No.         Time         Source         Destination         Protocol         Length         Priodol         Length         Priodol	tcp.port==1883			1
SCTP         TCP Stream         Ctd+At+Shift+T           0010         00 2 15 5d 20 44 81 02 15 5d 20 4         Follow         TCP Stream         Ctd+At+Shift+T           0010         00 2 07 5b 04 03 65 cd 5c 5d 7         Ocopy         UDP Stream         Ctd+At+Shift+T           0010         00 2 17 7b 04 03 5c cd 5c 5d 7         Ocopy         UDP Stream         Ctd+At+Shift+U           0010         00 2 16 c6 57 27         Description         TLS Stream         Ctd+At+Shift+S	0. Tme Source 100 137.463250 10.10.1.2 238 109.599717 10.10.1.12 240 109.540172 10.10.1.16 241 109.612071 10.10.1.16 241 109.612071 10.10.1.12 242 109.612437 10.10.1.12 243 109.612597 10.10.1.12 243 109.612597 10.10.1.12 243 109.612597 10.10.1.12 243 109.612597 10.10.1.12 245 109.612597 10.10.1.12 247 109.612597 10.10.1.12 248 109.612597 10.10.1.12 249 109.612597 10.10.1.12 249 109.612597 10.10.1.12 249 109.612597 10.10.1.12 250 100.112 250 100.112 2	Destination Protocol Leng 10.10.1.16 TCP 5 14.16.1.5 MTCP 5 Mark/Unmark Packet Ctrl-M Ignore/Unignore Packet Ctrl-M Set/Unset Time Reference Ctrl+T Packet Comment Ctrl+Shift=T Packet Comment Ctrl+Alt+C Edit Resolved Name Apply as Filter Prepare as Filter Conversation Filter	h 1nfo 4 1027 + 1083 [ACK] Seq=7 Ack=7 Win=8212 Len=0 4 1027 + 1083 [ACK] Seq=7 Ack=7 Win=8212 Len=0 Publish Ack (id=2) [High = 1027 [ACK] Seq=49 Ack=11 Win=8212 Len=0 Publish Release (id=2) Publish Release (id=2) Publish Release (id=2) =	
Control         C2         25         26         44         81         92         75         26         44         81         92         75         92         75         76 <t< td=""><td></td><td>SCTP +</td><td></td><td>_</td></t<>		SCTP +		_
FIDEDCOL PRETERINCES FLITTE Drawn Chil, Ab. Chile, M.	0010         02         15         5d         20         44         81         02         15         5d         20         4           0010         00         52         off         99         40         00         80         66         4f         7         50           0120         01         02         07         5b         04         03         65         cd         5c         5d         7d           0120         01         02         07         5b         04         03         65         cd         5c         5d         7d           0120         01         02         07         5b         04         03         65         cd         5c         5d         7d         7d         7d         6d         6d         6d         04         03         65         cd         04         03         2d         1d         6d         6d         04         03         2d         0d         0a         4d         0d         0d <t< td=""><td>Fallow  Copy Protocol Preferences</td><td>TCP Stream         Ctrl+Alt+Shift+T           UDP Stream         Ctrl+Alt+Shift+U           TLS Stream         Ctrl+Alt+Shift+S           HTTD Stream         Ctrl+Alt+Shift+I</td><td></td></t<>	Fallow  Copy Protocol Preferences	TCP Stream         Ctrl+Alt+Shift+T           UDP Stream         Ctrl+Alt+Shift+U           TLS Stream         Ctrl+Alt+Shift+S           HTTD Stream         Ctrl+Alt+Shift+I	
euso 48 69 67 68 20 54 65 66 70 65 7 Decode As HTTP/2 Stream Show Packet in New Window QUIC Stream	1030 48 09 07 68 20 34 65 68 70 65 7	Decode As Show Packet in New Window	HTTP/2 Stream QUIC Stream	





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.









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**.



11	a 🛛 🔅 🗋 🚍	XDQ .	2 ? <u>?</u> <u>.</u> <u>.</u> Q	Q Q II		
A to	p.stream eq 12				8	1 -
No.	Time	Source	Destination	Protocol	Length Info	-
	132 -488.728252	10.10.1.2	10.10.1.16	MQTT	56 Ping Request	
	133 -408.727726	10.10.1.16	10.10.1.2	MQTT	56 Ping Response	
	134 -408.487661	10.10.1.2	10.10.1.16	TCP	54 1027 - 1883 [ACK] Seq=5 Ack=5 Win=8212 Len=0	
	188 -350.780142	10.10.1.2	10.10.1.16	MQTT	56 Ping Request	
	189 -350.779667	10.10.1.16	10.10.1.2	MQTT	56 Ping Response	
	190 -350.690785	10.10.1.2	10.10.1.16	TCP	54 1027 - 1883 [ACK] Seq=7 Ack=7 Win=8212 Len=0	
	238 -298.594318	10.10.1.16	10.10.1.2	MQTT	96 Publish Message (id=2) [High_Tempe]	
FIET	thernet II, Src:	NS-NLB-PhysServer	-21 5d:20:44:84 (02:15	(768 bits) c	n interface \Device\NPf_{26C51D69-3344-487E-9225-4309C2E0338A}, id . Dst: MS-NLB-PhysServer-21_56226:44:81 (02:15:5d:20:44:81)	9 6
	thernet II, Src: nternet Protocol ransmission Contr Q Telemetry Trans	NS-NL8-PhysServer Version 4, Src: 1 Tol Protocol, Src sport Protocol, Pu	-21 Sd:28:44:84 (82:15 8-: 4 Unsaved packets Por b1: Do you want t Your captured	( 768 bits) c is5d:20:44:84) to save the captur I packets will be k	n interface UDeviceUNPF_(20C51009-3344-487E-9223-4399C288338A), id 	10
0000 0010 0010	thernet II, Src: nternet Protocol ransmission Contr Q Telemetry Trans 0 02 15 5d 20 44 0 00 52 0f 89 40 01 02 07 5b 49 01 02 07 5b 49 02 14 26 60	NS-NL8-PhysServer Version 4, Src: 1 rol Protocol, Src sport Protocol, Pu 81 02 15 5d 20 00 80 06 d4 17 03 65 cd 5c 5d	All and a set of the set of	to save the captur packets will be lo fore Continue	n inter-face UDevice/NPF_(20C51009-3344-487E-9225-4399C288338A), id . Dst:: MS-BLB.PBNusServer21_54:28:244:81 (02:15:55:28:44:81) × ed packets before restarting the capture? Int if you don't save them. Continue without Saving Cancel	10
P F F F F F F F F F F F F F F F F F F F	thernet II, Src: nternet Protocol ransmission Contr Q Telemetry Trans 0 02 15 5d 20 44 0 03 52 0f 89 46 01 02 07 5b 64 01 02 07 5b 64 05 6d 76 65 06	NS-NL8-PhysServer Version 4, Src: 1 ol Protocol, Src sport Protocol, Pu 08 08 06 04 77 03 65 cd 55 54 00 32 28 00 08 02 41 66 65 72	Si ), so bytes captures     Captures	(788 515) c is5d:28:44:84) to save the captur   packets will be k efore Continue (	n interface UDevice.VMPT_(20C51008-3344-487E-9225-4399C288338A), is 	4 0





- 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++.









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.



-Ci	\Bevywise\MQTTRoute\conf\broker.conf - Notepad++ [Administrator]	-	o x
Eile	<u>Edit Search View Encoding Language Settings Tools Macro Bun Plugins Window 2</u>		
6	) 🖶 🗞 🕼 🍐 X 🎕 🕼 🕽 🗢 C 🃾 🍢 🤏 📽 🖾 🖾 1 麗 🖼 🖉 💷 👁 🖲 🖲 🗎		
H host	ts 🖸 🔚 Web config 🕄 📕 ogin aspx cs 🕄 🔚 broker.conf 🖾		
11	# WS_PORT_NO - port to start the MQTT in Websocket		
12			
13	# TLS_ENABLED - If set us TRUE, the product will run with SSL and WSS.		
14	# TLS_PORT - The port at which the SSL version need to run.		
15	# WSS_PORT_NO - port to start the MQTT SSL version in Websocket		
16			
17	# PREFIX - Generate the random clientid with given prefix		
18			
19	**************************************		
20	[CONFIG]		
21			
22	PORT_NO = 1883		
23	WS_PORT_NO = 10443		
24			
25	TLS ENABLED = TRUE		
20	ILS_PORT must be doxx.		
21	ILS PORT NO = 5853		
20	ND5_PURI_DU = 11115		
20	And a second s		
30	(Authentication)		
32	[Notified Fort [ ]		
3.2	LITHENTICATION ENDERD = NO		
34	* YES II NO		
35			
36			
37	######################################		
38	[UI]		
39	UI Http Port = 8080		
40			
41	LIST API CLIENTS = FALSE		
42			
43			
44	########### prefix for Random Clientid Generation ####################################		
45	(MQTT]		
4.6			
47	CLIENTID_PREFIX = Bevywise-		
Normal	Literat File Length : 1.227 Lines : 47 Line : 25 Col : 10 Sel : 010 Windows (/	CRIED LITE-8	INC
a working the	Hindows (N	mary elline	192



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

EXERCISE 1: secure iot device communication using tls/ssl

File Home	Share	View	Application Tools Manage	bin					
← → • ↑	> This	PC > Loc	al Disk (C:) > Bevy	wise >	MQTTRoute > bin			~ (	5
		Name	^		Date modified	Туре	Size		
A Quick access		install	ler		8/1/2019 12:18 PM	Windows Batch File	9 KB		
Desktop	×	💿 runbr	oker		8/1/2019 12:18 PM	Windows Batch File	1 KB		
Downloads	*	💿 runga	steway		11/1/2018 4:56 PM	Windows Batch File	1 KB		
Documents	*	stopb	roker		8/1/2019 12:18 PM	Windows Batch File	1 KB		
Pictures	A								_





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.



C:\Windows\system32\cmd.exe

Sevywise MQTTRoute 2.0 - build 0719-030 Sevywise MQTTRoute - Trial Version - expires on Fri Sep 17 02:09:41 2021 TCP Port - 8883 WebSocket Port - 11443 //ew your connected devices via your browser at - http://localhost:8080





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.



🚺 🛛 🔁 🗖 🖛 🛛 Certi	ficate					_	
File Home	Share	View					
← → * ↑ 📘	> This	PC > Local Disk (C:) > Bevywise >	MQTTRoute > Certificate >			~	ō
- Ouick accare		Name	Date modified	Туре	Size		
Desiton		Client	8/18/2021 2:09 AM	File folder			
Desktop	~		8/18/2021 2:09 AM	File folder			
Downloads	*	server	8/18/2021 2:09 AM	File folder			
Documents	*						
Pictures	1						_





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

File Home	Module Share	View					
← → × ↑ 📙	> This P	C > CCT-Tools (\\ADMIN-MACHINE-1) (Z:)	CCT Module 13 loT a	and OT Security		~	õ
1 Orishaman		Name	Date modified	Туре	Size		
Cuick access		Bevywise IoT Simulator	6/27/2021 10:39 PM	File folder			
Desktop	×	client	8/18/2021 2:51 AM	File folder			
Downloads	*	root	8/18/2021 2:51 AM	File folder			
Documents	*						
E Pictures	*						







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









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

← → × ↑ 📙	« New	Volume (Z:) > CCT-Tools > CCT Mo	dule 13 IoT and OT Security >	~	Ö	P Search	CCT Module 13 IoT and 01
1 O.id.		Name	Date modified	Туре		Size	
Decites		Bevywise IoT Simulator	6/28/2021 1:39 AM	File folde	er		
Desktop		client	8/18/2021 5:51 AM	File folde	er		
Downloads	*	🧧 root	8/18/2021 5:51 AM	File folde	er		
CCT-Tools	*						





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

EXERCISE 1: Secure Iot Device communication using tls/ssl

> Local Disk (C:) > Bevywise >	lotSimulator > Certificate >	~	õ	, Search Certificate
ime ^	Date modified	Туре		Size
Authentication	8/18/2021 5:19 AM	File folde	r	
client	8/18/2021 5:53 AM	File folde	r	
root	8/18/2021 5:53 AM	File folde	t.	
server	8/18/2021 5:19 AM	File folde	r	
	> Local Disk (C:) > Bevywise > me Authentication Client root server	Local Disk (C:)      Bevywise      Interme     Date modified     Authentication     &/18/2021 5:19 AM     client     &/18/2021 5:53 AM     root     &/18/2021 5:53 AM     server     &/18/2021 5:19 AM	Local Disk (C:) > Bevywise > lotSimulator > Certificate >     Type     Date modified     Type     Authentication     8/18/2021 5:19 AM     File folde     teot     8/18/2021 5:53 AM     File folde     server     8/18/2021 5:53 AM     File folde     server	Local Disk (C:)      Bevywise      lotSimulator      Certificate      V     O      me     Date modified     Type     Authentication     8/18/2021 5:19 AM     File folder     toot     8/18/2021 5:33 AM     File folder     server     8/18/2021 5:19 AM     File folder





- 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\lotSimulator\bin** and double-click on the **runsimulator.bat** file.



1 🖸 🧧 🖬 1			Manage	bin			
File Home	Share	View	Application Tools				
← → ~ ↑ 📘	> This	PC > Loc	al Disk (C:) > Bevyw	ise > lotSimulator > bin	~	õ	,P Search bin
		Name	^	Date modified	Туре		Size
A Quick access		install	er	3/17/2020 8:53 PM	Windows	s Batch Fil	e 9 KB
Desktop	R	💿 runbr	oker	3/17/2020 8:48 PM	Windows	s Batch Fil	e 1 KB
Downloads	1	💿 runsir	nulator	3/17/2020 10:00 PM	Window:	s Batch Fil	e 1 KB
Documents	1						
CCT-Tools							
- New Volume	(D:)						
- New Volume	(F:)						





- 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.

ZXERCISE1: Secure Iot Device communication Using TLS/SSL









79. A Choose Network popup appears. Under Choose Network, select CCT\_FINANCE\_NETWORK and click on Open.



HEALTH_CARE ¥	HEALTH_CARE
HEALTH_CARE CCT_FINANCE_NETWORK	Description
	Includes Sensors and its details of a hospital
	Number of Devices
	12







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



imulator Settings			×
Broker Details Advan	ced		
Manager Application	Bevywise-IoT Platform	~	
Broker IP Address	10.10.1.16		
TLS/SSL	Enabled	~ Ø	
Broker Port	8883		
Root-Certificate:	Choose File No file chosen	upload	
Clean Session	0	0	
		Cancel	e





- 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.











- 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.**



♦ localhost.8080/login?next=%2F × +		0	-	σ	>
← → C ④ localhost8080/login?nex	t=%2F	Q,	\$	* 4	
(]) MQTTRoute					
	Sign in 💿				
	admin				
	Remember password				
	SIGN IN				





- 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**.



Bevywise MQT	TRoute - M	anage × +									0	-	(	3	×
← → C	<li>Iocalho</li>	ost:8080								04	Q	\$	*	-	:
() MQTTR	oute	Dashboard Devi	tes Topic	Message Rules	Error Log	Authent	ication	MQTT Clients	Tour			8		) Hel	p
Devices List	Q	Device Details		Connec	tion Status :0	Inline	Com	mand Send							
O TS1		Device Property	Va	lue			Topic				٦				
1		Client Name	TS	51			Hig	h_Tempe		~					
		From IP Address	10	10.1.2			This	is second alert fo	r High	÷					
		Connected On	18	Aug 2021 3:01:1	4		Terr	perature		10			_		
		WILL Topic	N	Ĺ									Se	nd- <b>a</b> l	
		WILL Message	N	Ĺ											
		WILL Retain	N	L											
		Messages Recei	ved											1	
		Topic		м	Message			QoS							
		High_Tempe		A	ert for High Te	emperatu	re		1-Atleast	Once					
		Subscribed Topic	:s											1	
	e 1 3	Subscribed Topic	5				QoS								
		High_Tempe					1-Adeas	st Once							





- 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.









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**.



	rc==10.10.1.16 && ip	.dst==10.10.1.2			
D.	Time	Source	Destination	Protocol L	ength Info
	58 26.786844	10.10.1.16	10.10.1.2	TCP	54 50475 + 445 [ACK] Seq=211 Ack=32981 Win=6176 Len=0
	59 26.786887	10.10.1.16	10.10.1.2	SM82	171 Read Request Len:28733 Off:30187520
	61 26.882102	10.10.1.16	10.10.1.2	TCP	54 58472 + 445 [ACK] Seq=587 Ack=29146 Win=8212 Len=8
	62 26.885514	10.10.1.16	10.10.1.2	SMB2	171 Read Request Len:32768 Off:45856
	64 26.977988	10.10.1.16	10.10.1.2	TCP	54 50473 + 445 [ACK] Seq=211 Ack=32981 Win=4283 Len=0
	65 26.977909	10.10.1.16	10.10.1.2	SM82	171 Read Request Len: 32768 Off: 77824 File: CCT Module 13 IoT and O
	67 27.087811	10.10.1.16	10.10.1.2	TCP	54 50474 + 445 [ACK] Seq=1039 Ack=33473 Win=8212 Len=0
	77 29.715383	10.10.1.16	10.10.1.2	\$482	126 Cancel Request
	88 29.821685	10.10.1.16	10.10.1.2	SMB2	146 Close Request
	81 29.878103	10.10.1.16	10.10.1.2	TCP	54 58472 + 445 [ACK] Seq=659 Ack=29222 Win=8212 Len=8
	84 29.941615	10.10.1.16	10.10.1.2	SMB2	126 Cancel Request
	86 30.037537	10.10.1.16	10.10.1.2	SM82	146 Close Request
	88 30.096925	10.10.1.16	10.10.1.2	TCP	54 50475 + 445 [ACK] Seq=375 Ack=33185 Win=6175 Len=0
	90 30.167215	10.10.1.16	10.10.1.2	SM82	126 Cancel Request
	91 30.221837	10.10.1.16	10.10.1.2	TCP	54 58472 + 445 [ACK] Seq=751 Ack=29358 Win=8211 Len=8
	93 30.274234	10.10.1.16	10.10.1.2	SM82	146 Close Request
	95 30.359951	10.10.1.16	10.10.1.2	SMB2	126 Cancel Request
	96 30.489999	10.10.1.16	10.10.1.2	TCP	54 58473 + 445 [ACK] Seg=375 Ack=33185 Win=7071 Len=0
Et In Tr	hernet II, Src: ternet Protocol ansmission Cont:	MS-NLB-PhysServer Version 4, Src: 16 rol Protocol, Src 1	-21_5d:20:44:84 (02:1 9.10.1.16, Dst: 10.10 Port: 50474, Dst Port	5:5d:20:44:84), 9.1.2 :: 445, Seq: 103	Dst: PK5-NLB-PhysServer-21_5d:20:44:81 (02:15:5d:20:44:81) ), Ack: 33473, Len: 0
	02 15 5d 20 44	81 02 15 5d 20 4	4 84 88 88 45 88	·] D····] D····E	
00	00 28 0f ec 40	00 80 06 d4 be 0	a 0a 01 10 0a 0a	(	
10	AL AS -F 3- AL	DD DC 91 72 74 1	7 DD DE E4 50 10	rt.n.p	
10	01 02 c5 2a 01 20 14 4d 12 00	00		- 50	









	p.src==10.10.1.16 && p.	dst==10.10.1.2					X =	1.
io.	Time	Source	Destination	Protocol	Length Info			
	775 273.644588	10.10.1.16	10.10.1.2	TCP	54 58473 → 44	5 [ACK]	Seq=4876 Ack=36015 Win=8033 Len=0	_
	798 274.144147	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seq=5298 Ack=36955 Win=8035 Len=0	
	818 274.501054	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seq=5614 Ack=37219 Win=8034 Len=0	
	825 274.644009	10.10.1.16	10.10.1.2	TCP	54 58473 → 44	5 [ACK]	Seq=5890 Ack=37303 Win=8406 Len=0	
	841 275.097151	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	S [ACK]	Seq=6230 Ack=37667 Win=8404 Len=0	
	855 275.472272	10.10.1.16	10.10.1.2	TCP	54 50473 + 44	S [ACK]	Seq=6436 Ack=39007 Win=8406 Len=0	
	870 275.863091	10.10.1.16	10.10.1.2	TCP	54 50473 + 44	S [ACK]	Seg=6784 Ack=39371 Win=8484 Len=8	
	886 276.238648	10.10.1.16	10.10.1.2	TCP	54 50473 → 44	S [ACK]	Seq=6876 Ack=39599 Win=8484 Len=8	
	137 43.625210	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seq=703 Ack=33261 Win=7071 Len=0	
	905 276.691897	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seq=7224 Ack=39963 Win=8402 Len=0	
	917 277.097130	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seq=7572 Ack=40327 Win=8401 Len=0	
	937 277.503339	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seq=7868 Ack=40595 Win=8406 Len=0	
	949 277.784531	10.10.1.16	10.10.1.2	TCP	54 50473 + 44	S [ACK]	Seq=8324 Ack=41099 Win=8404 Len=0	
	967 287.142251	10.10.1.16	10.10.1.2	TCP	54 50473 + 44	S [ACK]	Seg=8396 Ack=41171 Win=8484 Len=8	
	3494 870.359930	10.10.1.16	10.10.1.2	TCP	54 58473 → 44	S [ACK]	Seq=8616 Ack=41359 Win=8403 Len=0	
	3905 1029.013958	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	S [ACK]	Seg=8708 Ack=41487 Win=8402 Len=0	
	52 26.618140	10.10.1.16	10.10.1.2	TCP	54 58473 + 44	5 [ACK]	Seg=94 Ack=129 Win=4278 Len=8	
T	67 27.087811	10.10.1.16	10.10.1.2	TCP	54 58474 + 44	5 [ACK]	Seg=1039 Ack=33473 Win=8212 Len=0	





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



		and groups and and are			A CALIFORNIA
io.	Time	Source	Destination	Protocol L	Length Info
	14 8.927359	10.10.1.16	10.10.1.2	TCP	54 50475 → 445 [ACK] Seq=94 Ack=129 Win=6171 Len=0
	3873 1013.258	138 10.10.1.16	10.10.1.2	TCP	54 8883 + 3424 [ACK] Seq=2602 Ack=531 Win=2102016 Len=0
	3877 1013.313	687 10.10.1.16	10.10.1.2	TCP	54 8883 - 3424 [ACK] Seq=2635 Ack=597 Win=2102016 Len=0
	3460 854.8340	49 10.10.1.16	10.10.1.2	TCP	66 8883 + 3424 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS
	3469 854.8468	37 10.10.1.16	10.10.1.2	TLSv1.2	87 Application Data
	3471 854.8485	30 10.10.1.16	10.10.1.2	TLSv1.2	88 Application Data
	3701 914.0144	06 10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data
	3828 972.7587	13 10.10.1.16	10.10.1.2	TL5v1.2	85 Application Data
	3871 1013.207	473 10.10.1.16	10.10.1.2	TLSv1.2	140 Application Data
	3875 1013.258	760 10.10.1.16	10.10.1.2	TLSv1.2	87 Application Data
	3910 1030.686	697 10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data
	3964 1088.569	150 10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data
	4102 1146.617	206 10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data
	828 274.5812	59 10.10.1.16	10.10.1.2	DCERPC	330 Bind: call_id: 2, Fragment: Single, 3 context items: SRVSVC \
	77 29.71538	3 10.10.1.16	10.10.1.2	SMB2	126 Cancel Request
	84 29.94161	5 10.10.1.16	10.10.1.2	SMB2	126 Cancel Request
	90 30.16721	5 10.10.1.16	10.10.1.2	SMB2	126 Cancel Request
	95 30.35995	1 10.10.1.16	10.10.1.2	SM82	126 Cancel Request



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



	= 10.10.1.16 BB ip.0	dst==10.10.1.2				×
÷	Time	Source	Destination	Protocol I	Length Info	
1	4 8.927359	10.10.1.16	10.10.1.2	TCP	54 50475 + 445 [ACK] Seq=94 Ack=129 Win=6171 Len=0	
387	3 1013.258138	10.10.1.16	10.10.1.2	TCP	54 8883 → 3424 [ACK] Seq=2602 Ack=531 Win=2102016 Len=0	
387	7 1013.313687	10.10.1.16	10.10.1.2	TCP	54 8883 + 3424 [ACK] Seq=2635 Ack=597 Win=2102016 Len=0	
346	8 854.834049	10.10.1.16	10.10.1.2	TCP	66 8883 → 3424 [SYN, ACK] Seq=0 Ack=1 Win+8192 Len=0 MSS=	1460 WS
346	9 854.846837	10.10.1.16	10.10.1.2	TLSv1.2	87 Application Data	
347	1 854.848530	10.10.1.16	10.10.1.2	TLSv1.2	88 Application Data	
370	1 914.014406	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data	
382	8 972.758713	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data	
387	1 1013.207473	10.10.1.16	10.10.1.2	TLSv1.2	140 Application Data	
387	5 1013.258760	10.10.1.16	10.10.1.2	TLSv1.2	87 Application Data	
391	0 1030.686697	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data	
396	4 1088.569150	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data	
410	2 1146.617206	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data	
82	0 274.501259	10.10.1.16	10.10.1.2	DCERPC	330 Bind: call_id: 2, Fragment: Single, 3 context items: S	RVSVC 1
7	7 29.715383	10.10.1.16	10.10.1.2	SM82	126 Cancel Request	
8	4 29.941615	10.10.1.16	10.10.1.2	SMB2	126 Cancel Request	
9	0 30.167215	10.10.1.16	10.10.1.2	5M82	126 Cancel Request	
. 9	5 30.359951	10.10.1.16	10.10.1.2	SM82	126 Cancel Request	
Ether	met II, Src: P met Protocol V mission Contro sport Layer Sec	<pre>HS-NLB-PhysServer- /ersion 4, Src: 16 bl Protocol, Src F :urity war: inplication</pre>	21_5d:20:44:84 (02:15 ).10.1.16, Dst: 10.10. Nort: 8883, Dst Port:	:5d:20:44:84), 1.2 3424, Seq: 242	, Dst: MS-NLB-PhysServer-21_5d:20:44:81 (02:15:5d:20:44:81) 20, Ack: 436, Len: 34	
Trans		Application Data	(23)			
Trans Trans V Tu	Content Type: Version: TLS	1.2 (0x0303)				
Trans Trans V Tu	Content Type: Version: TLS Length: 29 Encrypted App	1.2 (0x0303) lication Data: 1c	2b86b67ec390fcea6738d	747e6521df79e8	8dd62f58b897_	
Trans Trans Y Tu	Content Type: Version: TLS Length: 29 Encrypted App	1.2 (0x0303)	2b86b67ec390fcea6738d	747e6521df79e8	8dd62f5eb897_	







p.src=	= 10. 10. 1. 16 && p.	dst==10.10.1.2					X	*
	Time	Source	Destination	Protocol I	ength Info	^		
14	8.927359	10.10.1.16	10.10.1.2	TCP	54 58475 + 445 [ACK	] Seq=94 Ack=129 Win=6171 Len=0		
3873	1013.258138	10.10.1.16	10.10.1.2	TCP	54 8883 + 3424 [ACK	] Seq=2602 Ack=531 Win=2102016 Len=0		
3877	1013.313687	10.10.1.16	10.10.1.2	TCP	54 8883 - 3424 [ACK	] Seq=2635 Ack=597 Win=2102016 Len=0		
3468	854.834049	10.10.1.16	10.10.1.2	TCP	66 8883 + 3424 [SYN	, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS	=1468 WS	ŝ
3469	854.846837	10.10.1.16	10.10.1.2	TLSv1.2	87 Application Data			ĺ
3471	854.848530	10.10.1.16	10.10.1.2	TLSv1.2	88 Application Data			ĺ
3701	914.014406	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data			
3828	972.758713	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data			
3871	1013.207473	10.10.1.16	10.10.1.2	TLSv1.2	140 Application Data			
3875	1013.258760	10.10.1.16	10.10.1.2	TLSv1.2	87 Application Data			
3910	1030.686697	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data			
3964	1088.569150	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data			
4102	1146.617206	10.10.1.16	10.10.1.2	TLSv1.2	85 Application Data			
820	274.501259	10.10.1.16	10.10.1.2	DCERPC	330 Bind: call_id: 2	, Fragment: Single, 3 context items:	SRVSVC V	9
77	29.715383	10.10.1.16	10.10.1.2	SM82	126 Cancel Request			
84	29.941615	10.10.1.16	10.10.1.2	SM82	126 Cancel Request			
94	30.167215	10.10.1.16	10.10.1.2	\$482	126 Cancel Request			
- 95	30.359951	10.10.1.16	10.10.1.2	SMB2	126 Cancel Request			
Ethen Inter Trans Trans Y TL	net II, Src: 1 net Protocol 1 mission Contro port Layer Se Sv1.2 Record 1 Content Type: Version: TLS	NS-NLB-PhysServer-2 Version 4, Src: 10. ol Protocol, Src Pc curity layer: Application : Application Data 1.2 (0x0303)	11_50:20:44:84 (02:15 10.1.16, Dst: 10.10. prt: 8883, Dst Port: Data Protocol: mgtt (23)	:5d:20:44:84), 1.2 3424, Seq: 242	Dst: MS-NLB-PhysServ	er-21_5d:20:44:81 (02:15:5d:20:44:81)		
	Length: 29	lication Datas 1c2	hashered and the second stand	747-55 31 4570-0	and the second se			





- 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.



# EC-Council