Module 18: IoT and OT Hacking

Scenario

The significant development of the paradigm of the Internet of Things (IoT) is contributing to the proliferation of devices in daily life. From smart homes to automated healthcare applications, IoT is ubiquitous. However, despite the potential of IoT to make our lives easier and more comfortable, we cannot underestimate its vulnerability to cyber-attacks. IoT devices lack basic security, which makes them prone to various cyber-attacks.

The objective of a hacker in exploiting IoT devices is to gain unauthorized access to users' devices and data. A hacker can use compromised IoT devices to build an army of botnets, which, in turn, is used to launch DDoS attacks.

Owing to a lack of security policies, smart devices are easy targets for hackers who can compromise these devices to spy on users' activities, misuse sensitive information (such as patients' health records, etc.), install ransomware to block access to the devices, monitor victims' activities using CCTV cameras, commit credit-card-related fraud, gain access to users' homes, or recruit the devices in an army of botnets to carry out DDoS attacks.

As an ethical hacker and penetration tester, you must have sound knowledge of hacking IoT and OT platforms using various tools and techniques. The labs in this module will provide you with real-time experience in performing footprinting and analyzing traffic between IoT and OT devices.

Objective

The objective of the lab is to perform IoT and OT platform hacking and other tasks that include, but are not limited to:

- Performing IoT and OT device footprinting
- Capturing and analyzing traffic between IoT devices
- Performing IoT attacks

Overview of IoT and OT Hacking

Using the IoT and OT hacking methodology, an attacker acquires information using techniques such as information gathering, attack surface area identification, and vulnerability scanning, and uses such information to hack the target device and network.

The following are the various phases of IoT and OT device hacking:

- Information gathering
- Vulnerability scanning
- Launch attacks
- Gain remote access
- Maintain access

Lab Tasks

Ethical hackers or pen testers use numerous tools and techniques to hack the target IoT and OT platforms. Recommended labs that will assist you in learning various IoT platform hacking techniques include:

1. Perform footprinting using various footprinting techniques
    - Gather information using online footprinting tools
2. Capture and analyze IoT device traffic
    - Capture and analyze IoT traffic using Wireshark
3. Perform IoT Attacks
    - Perform replay attack on CAN protocol

Lab 1: Perform Footprinting using Various Footprinting Techniques

Lab Scenario

As a professional ethical hacker or pen tester, your first step is to gather maximum information about the target IoT and OT devices by performing footprinting through search engines, advanced Google hacking, Whois lookup, etc.

The first step in IoT and OT device hacking is to extract information such as IP address, protocols used (MQTT, ModBus, ZigBee, BLE, 5G, IPv6LoWPAN, etc.), open ports, device type, geolocation of the device, manufacturing number, and manufacturer of the device.

Lab Objectives

- Gather information using online footprinting tools

Overview of Footprinting Techniques

Footprinting techniques are used to collect basic information about the target IoT and OT platforms to exploit them. Information collected through footprinting techniques includes IP address, hostname, ISP, device location, banner of the target IoT device, FCC ID information, certification granted to the device, etc.

Task 1: Gather Information using Online Footprinting Tools

The information regarding the target IoT and OT devices can be acquired using various online sources such as Whois domain lookup, advanced Google hacking, and Shodan search engine. The gathered information can be used to scan the devices for vulnerabilities and further exploit them to launch attacks.

In this task, we will focus on performing footprinting on the MQTT protocol, which is a machine-to-machine (M2M)/“Internet of Things” connectivity protocol. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

You can also select a protocol or device of your choice to perform footprinting on it.

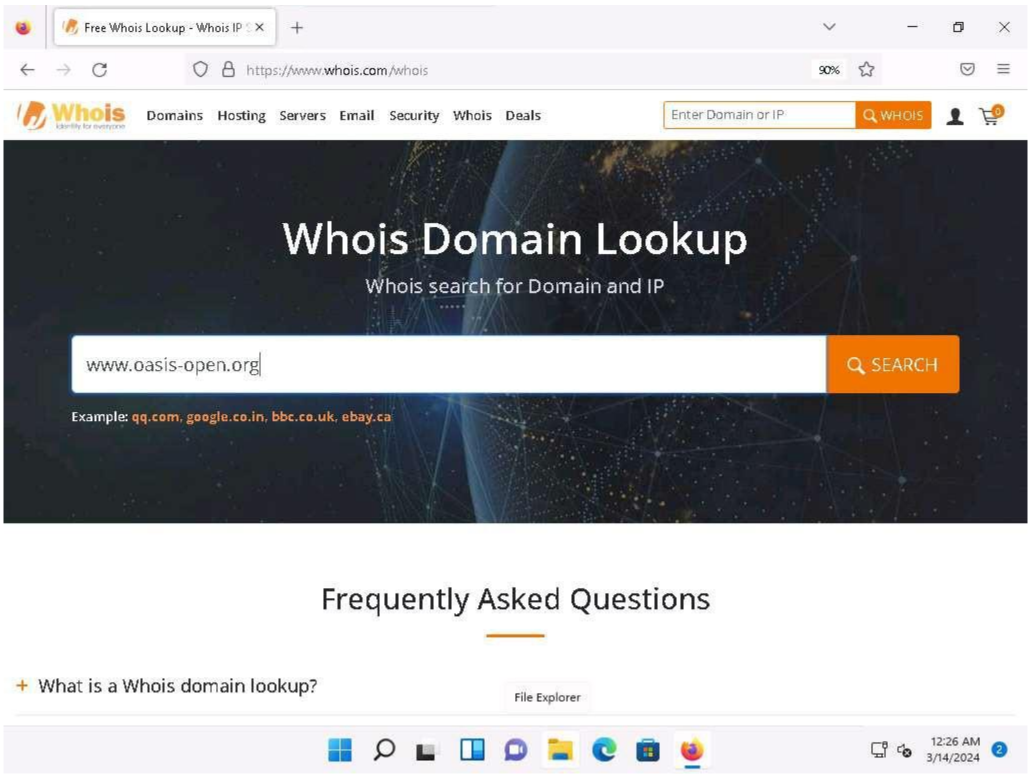
1. By default **Windows 11** machine selected, click Ctrl+Alt+Delete. Login with **Admin/Pa$$w0rd**.

   Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.

2. Launch any web browser, go to **https://www.whois.com/whois** (here, we are using **Mozilla Firefox**).

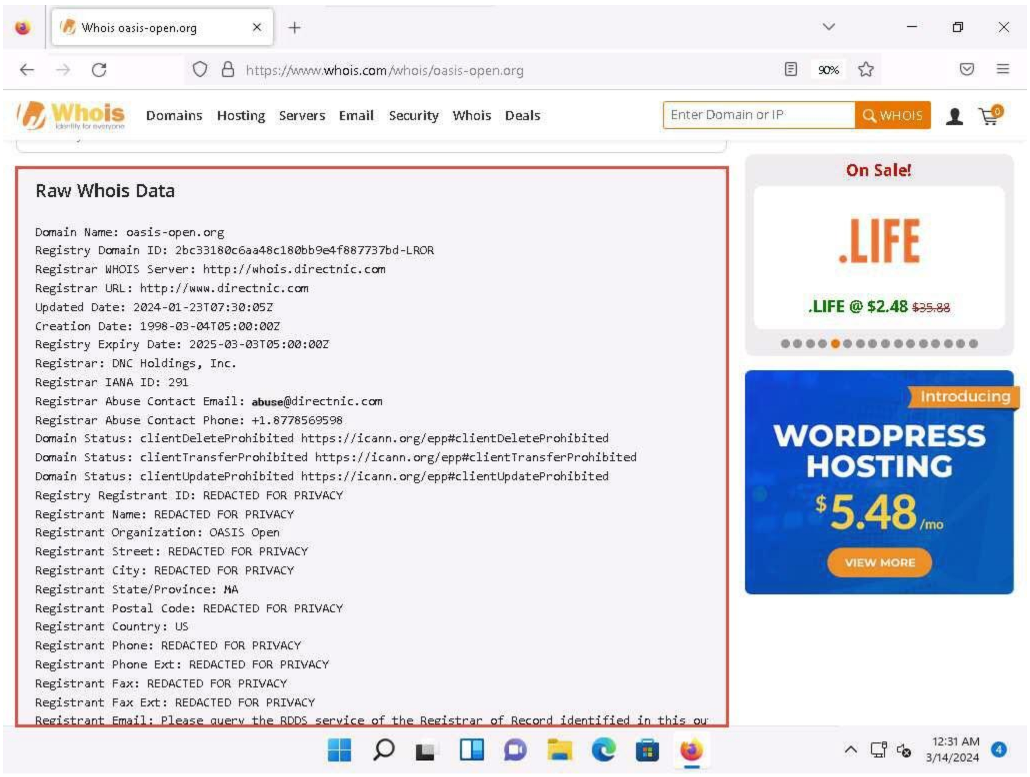
3. The **Whois Domain Lookup** page appears; type **www.oasis-open.org** in the search field and click **SEARCH**.

   Oasis is an organization that has published the MQTT v5.0 standard, which represents a significant leap in the refinement and capability of the messaging protocol that already powers IoT.


Free Whois Lookup - Whois IP
https://www.whois.com/whois
Whois
Domains Hosting Servers Email Security Whois Deals
Enter Domain or IP
WHOIS
Whois Domain Lookup
Whois search for Domain and IP
www.oasis-open.org
SEARCH
Example: qq.com, google.co.in, bbc.co.uk, ebay.ca
Frequently Asked Questions
+ What is a Whois domain lookup?
File Explorer
12:26 AM
3/14/2024

4. The result appears, displaying the following information, as shown in the screenshots: Domain Information, Registrant Contact, and Raw Whois Data.

   This information is about the organization that has developed the MQTT protocol, and it might help keep track of the modifications and version changes of the target protocol.



Raw Whois Data

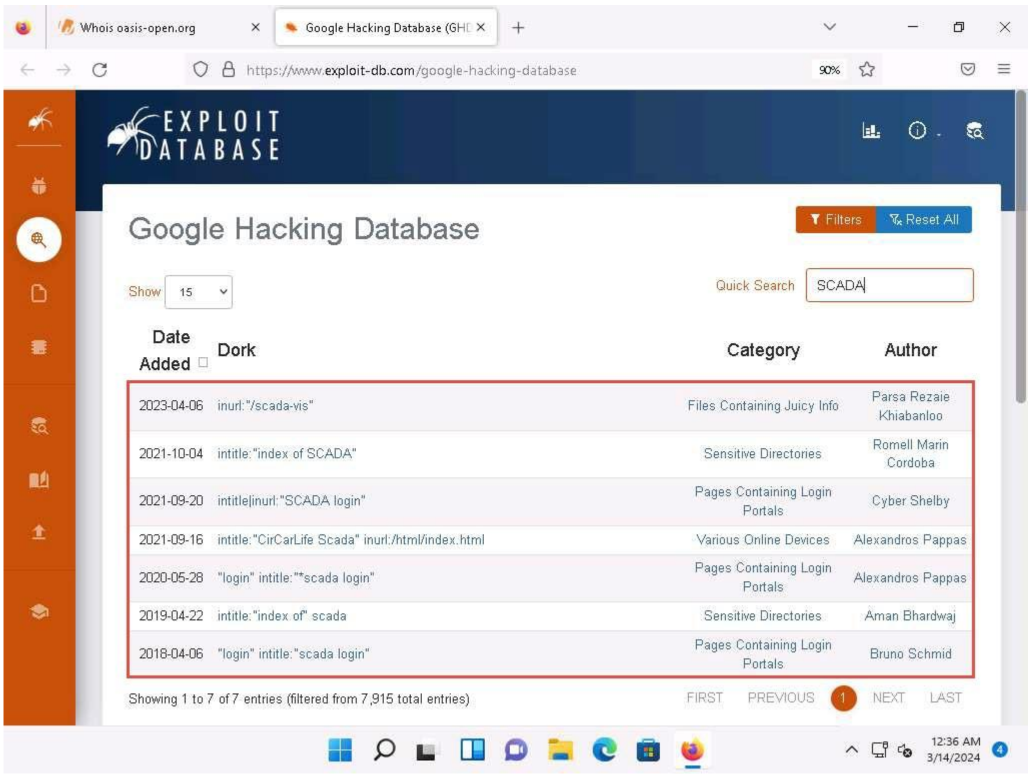
```
Domain Name: oasis-open.org
Registry Domain ID: 2bc33180c6aa48c180bb9e4f887737bd-LROR
Registrar WHOIS Server: http://whois.directnic.com
Registrar URL: http://www.directnic.com
Updated Date: 2024-01-23T07:30:05Z
Creation Date: 1998-03-04T05:00:00Z
Registry Expiry Date: 2025-03-03T05:00:00Z
Registrar: DNC Holdings, Inc.
Registrar IANA ID: 291
Registrar Abuse Contact Email: abuse@directnic.com
Registrar Abuse Contact Phone: +1.8778569598
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: OASIS Open
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: MA
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: US
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext: REDACTED FOR PRIVACY
Registrant Email: Please query the RDDS service of the Registrar of Record identified in this ou
```

   Whois lookup reveals available information on a hostname, IP address, or domain.

5. Now, open a new tab, and go to **https://www.exploit-db.com/google-hacking-database**.

6. The **Google Hacking Database** page appears; type **SCADA** in the **Quick Search** field and press **Enter**.

7. The result appears, which displays the Google dork related to SCADA, as shown in the screenshot.


Google Hacking Database

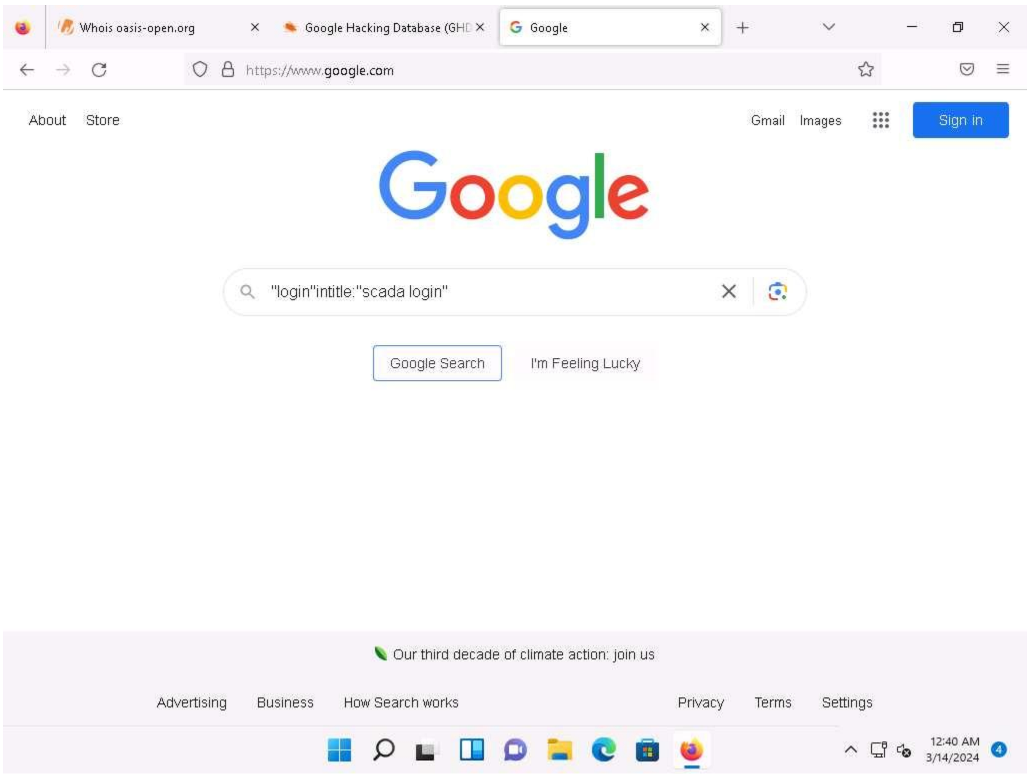
Show 15 Quick Search SCADA

Date Added	Dork	Category	Author
2023-04-06	inurl:"/scada-vis"	Files Containing Juicy Info	Parsa Rezaie Khiabanloo
2021-10-04	intitle:"index of SCADA"	Sensitive Directories	Romell Marin Cordoba
2021-09-20	intitle|inurl:"SCADA login"	Pages Containing Login Portals	Cyber Shelby
2021-09-16	intitle:"CirCarLife Scada" inurl:/html/index.html	Various Online Devices	Alexandros Pappas
2020-05-28	"login" intitle:"*scada login"	Pages Containing Login Portals	Alexandros Pappas
2019-04-22	intitle:"index of" scada	Sensitive Directories	Aman Bhardwaj
2018-04-06	"login" intitle:"scada login"	Pages Containing Login Portals	Bruno Schmid

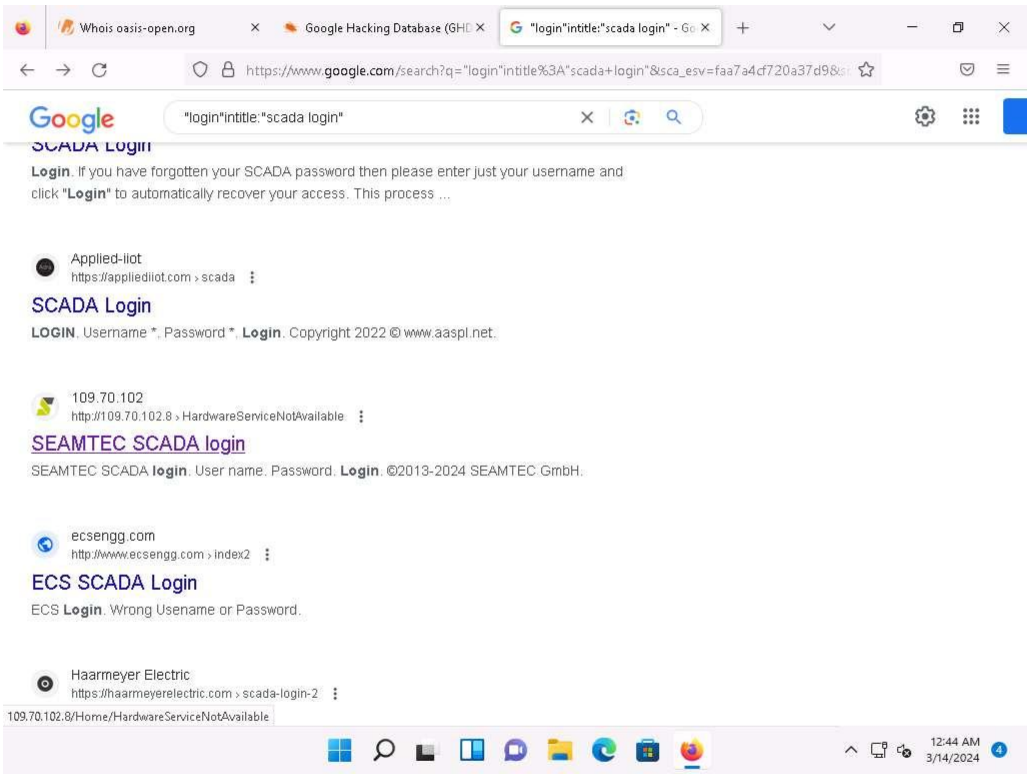
Showing 1 to 7 of 7 entries (filtered from 7,915 total entries)

8. Now, we will use the dorks obtained in the previous step to query results in Google.

9. Open a new tab and go to **https://www.google.com**. In the search field, enter **"login" intitle:"scada login"**.



10. The search result appears; click any link (here, **SEAMTEC SCADA login**).



Whois oasis-open.org

Google Hacking Database (GHD

"login"intitle:"scada login" - Go

https://www.google.com/search?q="login"intitle%3A"scada+login"&sca_esv=faa7a4cf720a37d9&s

"login"intitle:"scada login"

SCADA Login

Login. If you have forgotten your SCADA password then please enter just your username and click "**Login**" to automatically recover your access. This process ...

Applied-iiot
https://appliediiot.com › scada

SCADA Login

LOGIN. Username *. Password *. **Login**. Copyright 2022 © www.aaspl.net.

109.70.102
http://109.70.102.8 › HardwareServiceNotAvailable

SEAMTEC SCADA login

SEAMTEC SCADA **login**. User name. Password. **Login**. ©2013-2024 SEAMTEC GmbH.

ecsengg.com
http://www.ecsengg.com › index2

ECS SCADA Login

ECS **Login**. Wrong Usename or Password.

Haarmeyer Electric
https://haarmeyerelectric.com › scada-login-2

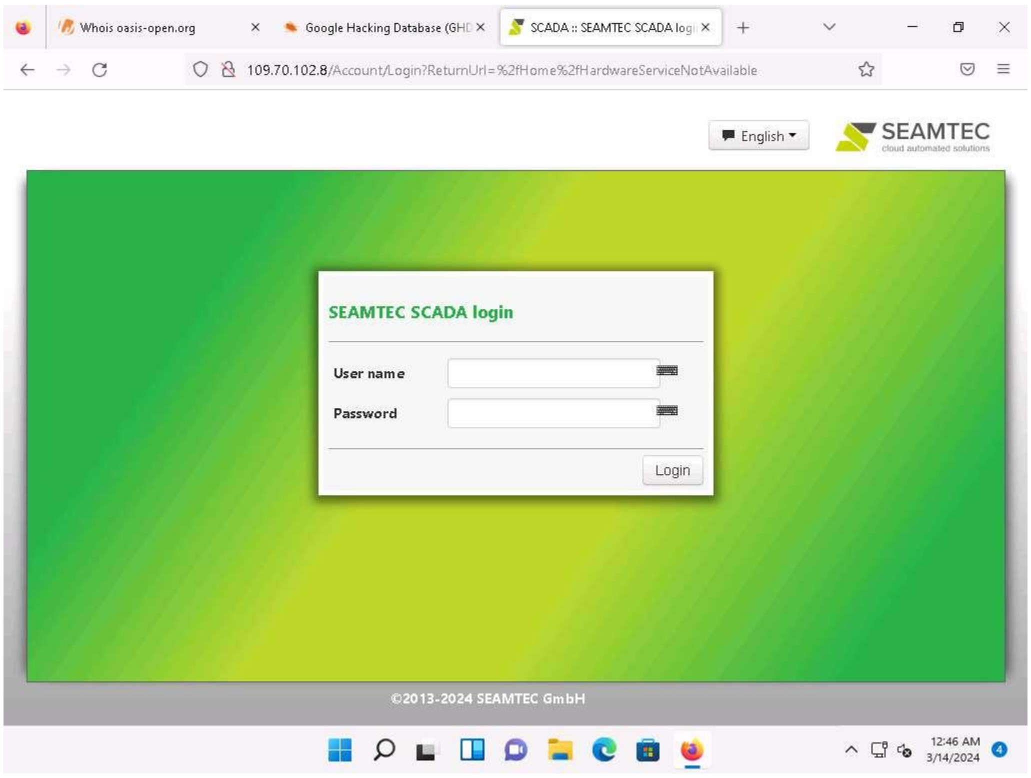
109.70.102.8/Home/HardwareServiceNotAvailable

12:44 AM
3/14/2024

Advanced Google hacking refers to the art of creating complex search engine queries by employing advanced Google operators to extract sensitive or hidden information about a target company from the Google search results.

11. The **SEAMTEC SCADA login** page appears, as shown in the screenshot.

In the login form, you can brute-force the credentials to gain access to the target SCADA system.

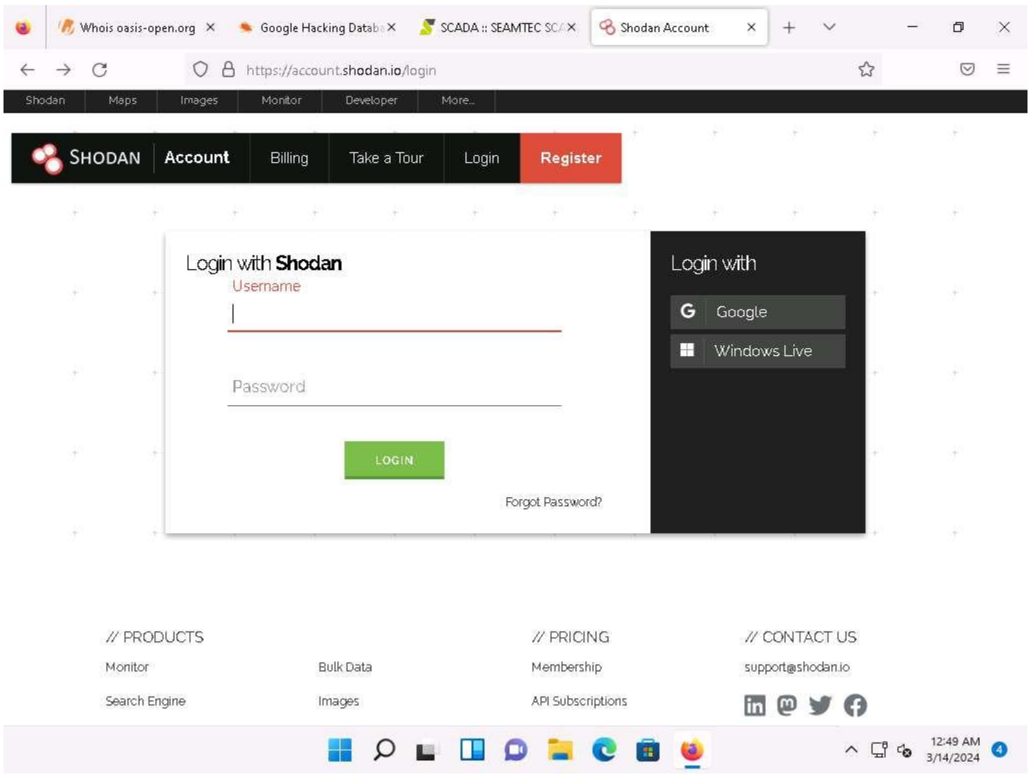

Whois oasis-open.org
Google Hacking Database (GHD
SCADA :: SEAMTEC SCADA logi
109.70.102.8/Account/Login?ReturnUrl=%2fHome%2fHardwareServiceNotAvailable
English
SEAMTEC
cloud automated solutions
SEAMTEC SCADA login
User name
Password
Login
©2013-2024 SEAMTEC GmbH
12:46 AM
3/14/2024

12. Similarly, you can use advanced search operators such as **intitle:"index of" scada** to search sensitive SCADA directories that are exposed on sites.

13. Now, in the browser window, open a new tab and go to **https://account.shodan.io/login**.

14. The **Login with Shodan** page appears; enter your username and password in the **Username** and **Password** fields, respectively; and click **Login**.

    If you do not have an existing account, then go to the **Register** option to register yourself .

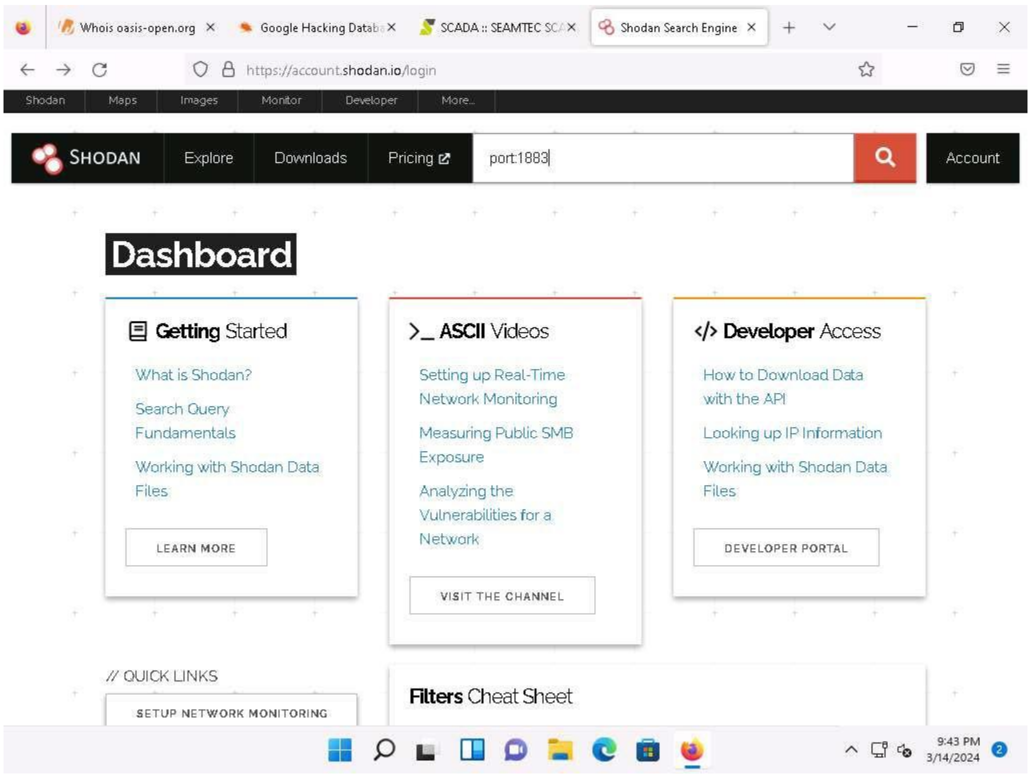


15. The **Account Overview** page appears, which displays the account-related information. Click on **Shodan** on top-left corner of the window to go to the main page of **Shodan**.

    If the **Would you like Firefox to save this login for shodan.io?** notification appears, click **Don't Save**.

16. The **Shodan** main page appears; type **port:1883** in the address bar and press **Enter**.

    Port 1883 is the default MQTT port; 1883 is defined by IANA as MQTT over TCP.



Dashboard

Getting Started
What is Shodan?
Search Query Fundamentals
Working with Shodan Data Files
LEARN MORE

>_ ASCII Videos
Setting up Real-Time Network Monitoring
Measuring Public SMB Exposure
Analyzing the Vulnerabilities for a Network
VISIT THE CHANNEL

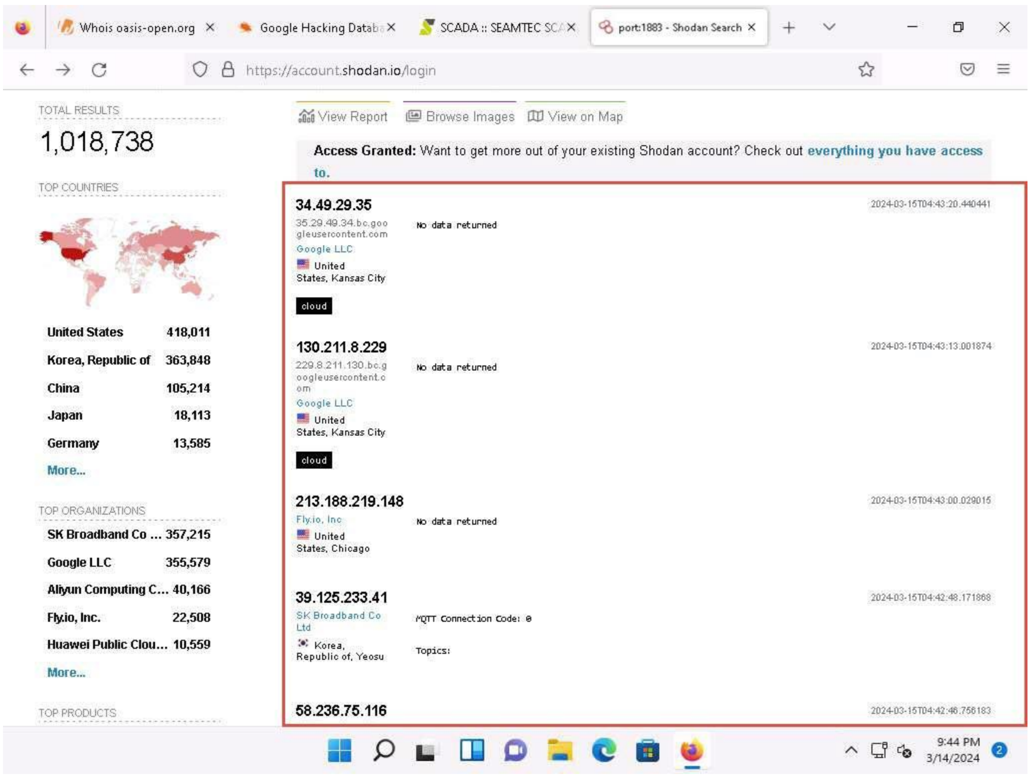
</> Developer Access
How to Download Data with the API
Looking up IP Information
Working with Shodan Data Files
DEVELOPER PORTAL

// QUICK LINKS
SETUP NETWORK MONITORING

Filters Cheat Sheet

17. The result appears, displaying the list of IP addresses having port 1883 enabled.

18. Click on any IP address to view its detailed information.



TOTAL RESULTS
1,018,738

TOP COUNTRIES

United States	418,011
Korea, Republic of	363,848
China	105,214
Japan	18,113
Germany	13,585
More...	

TOP ORGANIZATIONS

SK Broadband Co ...	357,215
Google LLC	355,579
Aliyun Computing C...	40,166
Fly.io, Inc.	22,508
Huawei Public Clou...	10,559
More...	

TOP PRODUCTS

Access Granted: Want to get more out of your existing Shodan account? Check out **everything you have access to.**

34.49.29.35 2024-03-15T04:43:20.440441
35.29.49.34.bc.googleusercontent.com
Google LLC
United States, Kansas City
cloud
```
No data returned
```

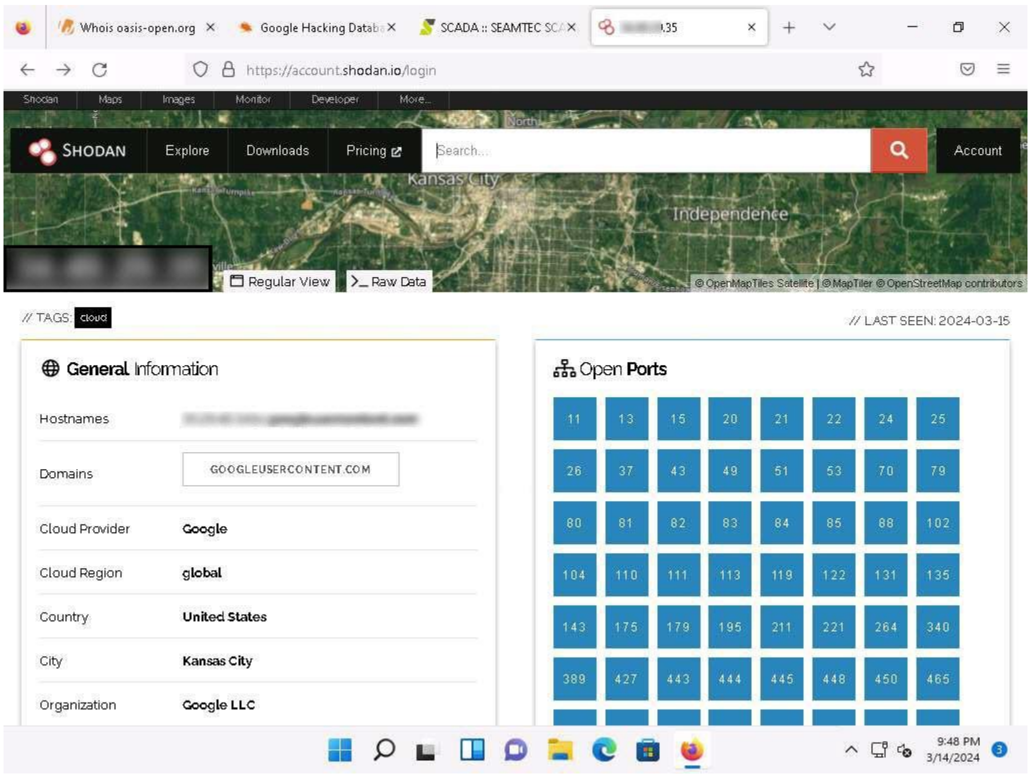
130.211.8.229 2024-03-15T04:43:13.001874
229.8.211.130.bc.googleusercontent.com
Google LLC
United States, Kansas City
cloud
```
No data returned
```

213.188.219.148 2024-03-15T04:43:00.029015
Fly.io, Inc
United States, Chicago
```
No data returned
```

39.125.233.41 2024-03-15T04:42:48.171868
SK Broadband Co Ltd
Korea, Republic of, Yeosu
```
MQTT Connection Code: 0

Topics:
```

58.236.75.116 2024-03-15T04:42:46.758183

19. Detailed results for the selected IP address appears, displaying information regarding **Ports, Services, Hostnames, ASN**, etc. as shown in the screenshot.



[redacted]

// TAGS: cloud

// LAST SEEN: 2024-03-15

General Information

Hostnames	[redacted]
Domains	GOOGLEUSERCONTENT.COM
Cloud Provider	Google
Cloud Region	global
Country	United States
City	Kansas City
Organization	Google LLC

Open Ports

11	13	15	20	21	22	24	25
26	37	43	49	51	53	70	79
80	81	82	83	84	85	88	102
104	110	111	113	119	122	131	135
143	175	179	195	211	221	264	340
389	427	443	444	445	448	450	465

20. Similarly, you can gather additional information on a target device using the following Shodan filters:

    - **Search for Modbus-enabled ICS/SCADA systems:**

      port:502

    - **Search for SCADA systems using PLC name:**

      “Schneider Electric”

    - **Search for SCADA systems using geolocation:**

      SCADA Country:"US"

21. Using Shodan, you can obtain the details of SCADA systems that are used in water treatment plants, nuclear power plants, HVAC systems, electrical transmission systems, home heating systems, etc.

22. This concludes the demonstration of gathering information on a target device using various techniques such as Whois lookup, advanced Google hacking, and Shodan search engine.

23. Close all open windows and document all the acquired information.

Question 18.1.1.1

Use the Shodan search engine to collect the IP addresses with MQTT enabled. Perform a search using the MQTT port number. Which port number will you enter in the search field to obtain the desired result?

Lab 2: Capture and Analyze IoT Device Traffic

Lab Scenario

As a professional ethical hacker or pen tester, you must have sound knowledge to capture and analyze the traffic between IoT devices. Using various tools and techniques, you can capture the valuable data flowing between the IoT devices, analyze it to obtain information on the communication protocol used by the IoT devices, and acquire sensitive information such as credentials, device identification numbers, etc.

Lab Objectives

- Capture and analyze IoT traffic using Wireshark

Overview of IoT and OT Traffic

Many IoT devices such as security cameras host websites for controlling or configuring cameras from remote locations. These websites mostly implement the insecure HTTP protocol instead of the secure HTTPS protocol and are, hence, vulnerable to various attacks. If the cameras use the default factory credentials, an attacker can easily intercept all the traffic flowing between the camera and web applications and further gain access to the camera itself. Attackers can use tools such as Wireshark to intercept such traffic and decrypt the Wi-Fi keys of the target network.

Task 1: Capture and Analyze IoT Traffic using Wireshark

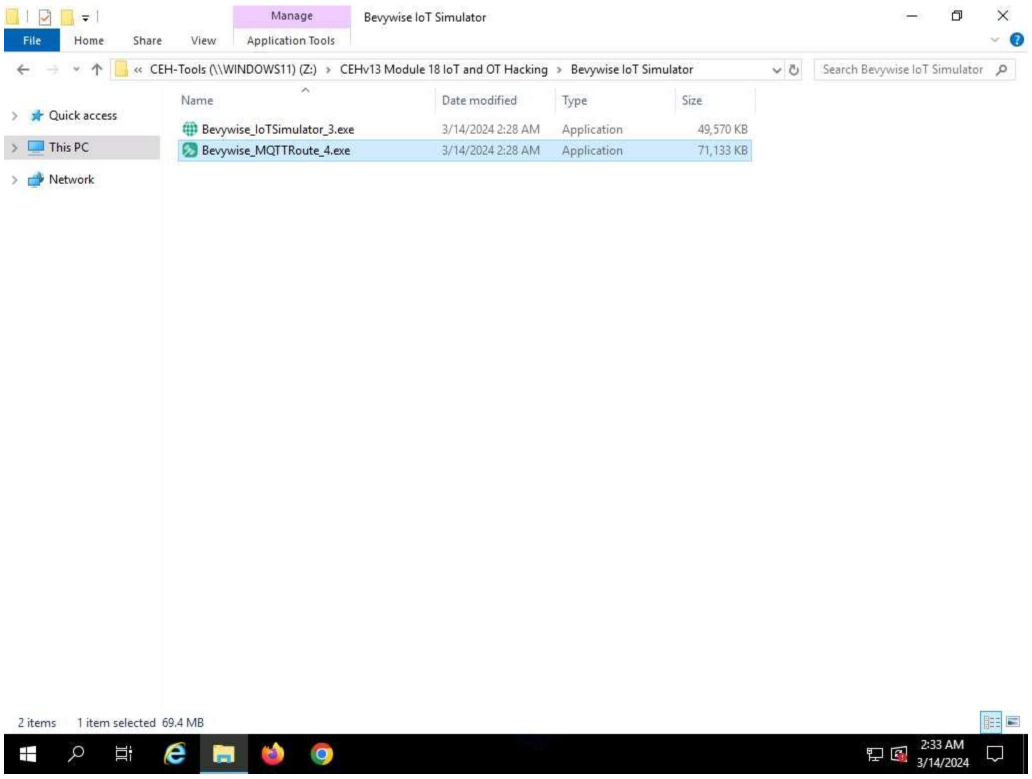
Wireshark is a free and open-source packet analyzer. It facilitates network troubleshooting, analysis, software and communications protocol development, and education. It is used to identify the target OS and sniff/capture the response generated from the target machine to the machine from which a request originates.

MQTT is a lightweight messaging protocol that uses a publish/subscribe communication pattern. Since 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 connected devices over the network.

Here, we use Wireshark to capture and analyze traffic between IoT devices.

1. To install the **MQTT Broker** on the **Windows Server 2019**, click Windows Server 2019 to launch **Windows Server 2019** machine.
   Click Ctrl+Alt+Delete and login with **Administrator/Pa$$w0rd**.

2. Navigate to **Z:\CEHv13 Module 18 IoT and OT Hacking\Bevywise IoT Simulator** folder and double-click on the **Bevywise_MQTTRoute_4.exe** file.

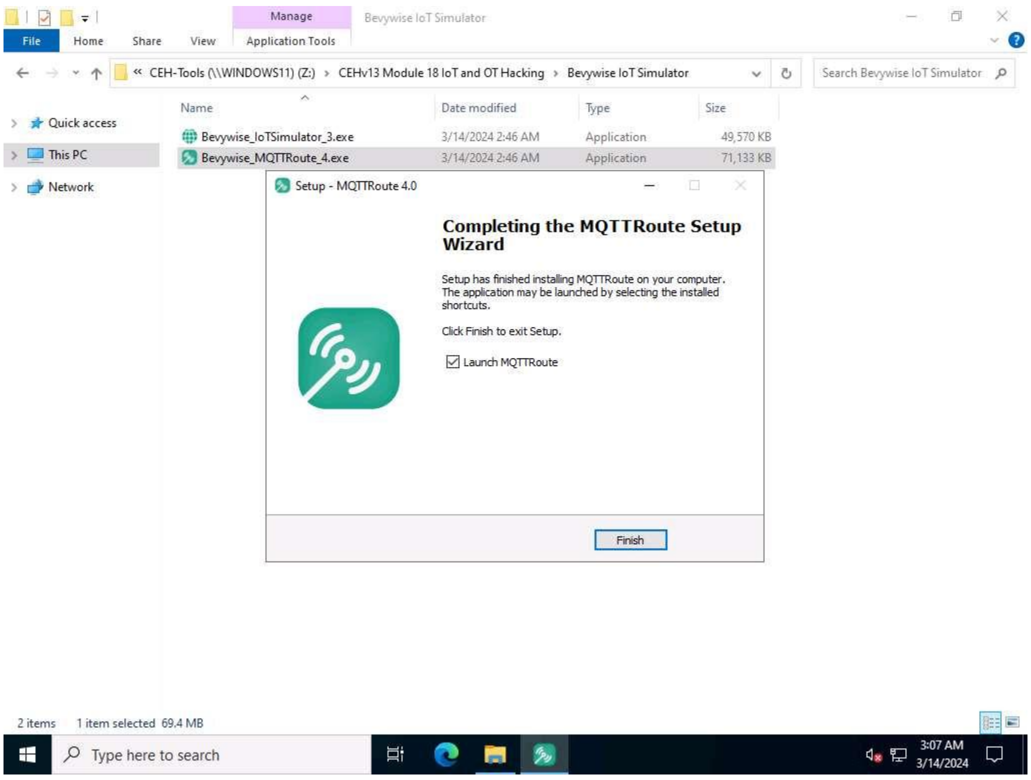

Manage
Bevywise IoT Simulator
File
Home
Share
View
Application Tools
« CEH-Tools (\\WINDOWS11) (Z:) › CEHv13 Module 18 IoT and OT Hacking › Bevywise IoT Simulator
Search Bevywise IoT Simulator
Quick access
This PC
Network
2 items   1 item selected 69.4 MB
2:33 AM
3/14/2024

Name	Date modified	Type	Size
Bevywise_IoTSimulator_3.exe	3/14/2024 2:28 AM	Application	49,570 KB
Bevywise_MQTTRoute_4.exe	3/14/2024 2:28 AM	Application	71,133 KB

3. If **Open File - Security Warning** popup appears, click **Run**.

4. The **Setup – MQTTRoute 4.0** window opens. Select **I accept the agreement** and click on **Next**. Follow the wizard driven steps to install the tool.

5. After the installation completes, click on **Finish**. Ensure that **Launch MQTTRoute** is checked.



Name	Date modified	Type	Size
Bevywise_IoTSimulator_3.exe	3/14/2024 2:46 AM	Application	49,570 KB
Bevywise_MQTTRoute_4.exe	3/14/2024 2:46 AM	Application	71,133 KB

Setup - MQTTRoute 4.0

Completing the MQTTRoute Setup Wizard

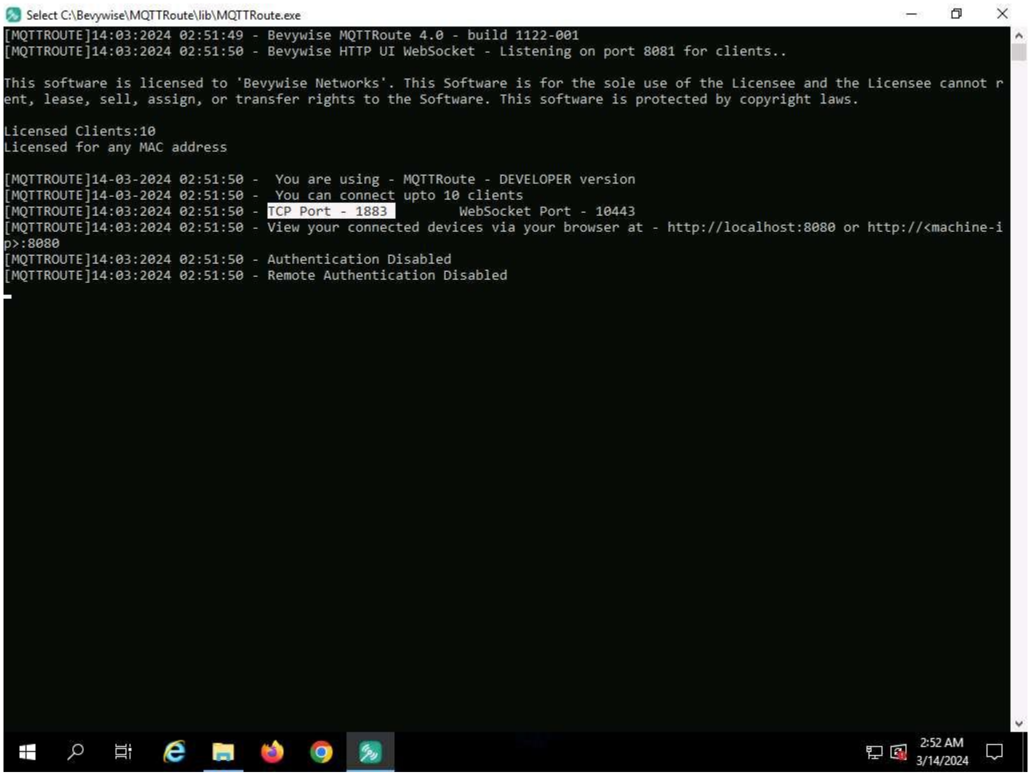
Setup has finished installing MQTTRoute on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

[✓] Launch MQTTRoute

Finish

6. The MQTTRoute will execute and the command prompt will appear. You can see the **TCP** port using **1883**.



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

```
[MQTTROUTE]14:03:2024 02:51:49 - Bevywise MQTTRoute 4.0 - build 1122-001
[MQTTROUTE]14:03:2024 02:51:50 - Bevywise HTTP UI WebSocket - Listening on port 8081 for clients..

This software is licensed to 'Bevywise Networks'. This Software is for the sole use of the Licensee and the Licensee cannot r
ent, lease, sell, assign, or transfer rights to the Software. This software is protected by copyright laws.

Licensed Clients:10
Licensed for any MAC address

[MQTTROUTE]14-03-2024 02:51:50 -  You are using - MQTTRoute - DEVELOPER version
[MQTTROUTE]14-03-2024 02:51:50 -  You can connect upto 10 clients
[MQTTROUTE]14:03:2024 02:51:50 - TCP Port - 1883         WebSocket Port - 10443
[MQTTROUTE]14:03:2024 02:51:50 - View your connected devices via your browser at - http://localhost:8080 or http://<machine-i
p>:8080
[MQTTROUTE]14:03:2024 02:51:50 - Authentication Disabled
[MQTTROUTE]14:03:2024 02:51:50 - Remote Authentication Disabled
```

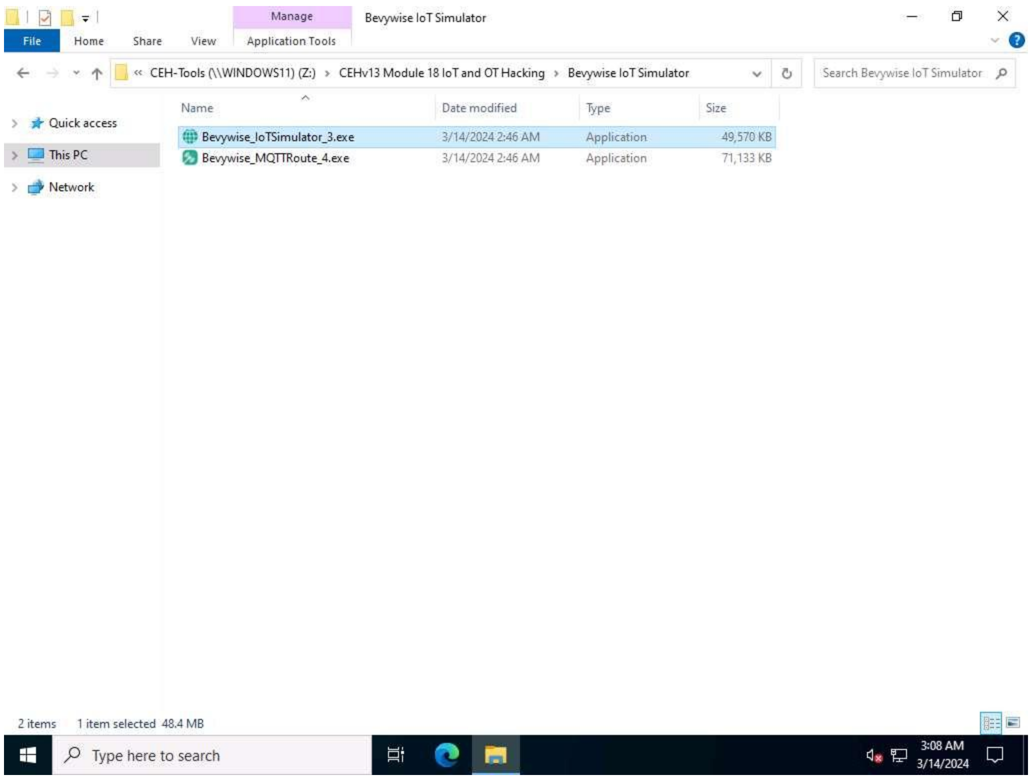
7. We have installed MQTT Broker successfully and leave the Bevywise MQTT **running**.

8. To create IoT devices, we must install the **IoT simulator** on the client machine.

9. Click Windows Server 2022 to switch to **Windows Server 2022** machine. Click Ctrl+Alt+Delete and login with **Administrator/Pa$$w0rd**.

   If the network screen appears, click **Yes**.

10. Navigate to **Z:\CEHv13 Module 18 IoT and OT Hacking\Bevywise IoT Simulator** folder and double-click on the **Bevywise_IoTSimulator_3.exe** file.



Bevywise IoT Simulator

CEH-Tools (\\WINDOWS11) (Z:) > CEHv13 Module 18 IoT and OT Hacking > Bevywise IoT Simulator

Name	Date modified	Type	Size
Bevywise_IoTSimulator_3.exe	3/14/2024 2:46 AM	Application	49,570 KB
Bevywise_MQTTRoute_4.exe	3/14/2024 2:46 AM	Application	71,133 KB

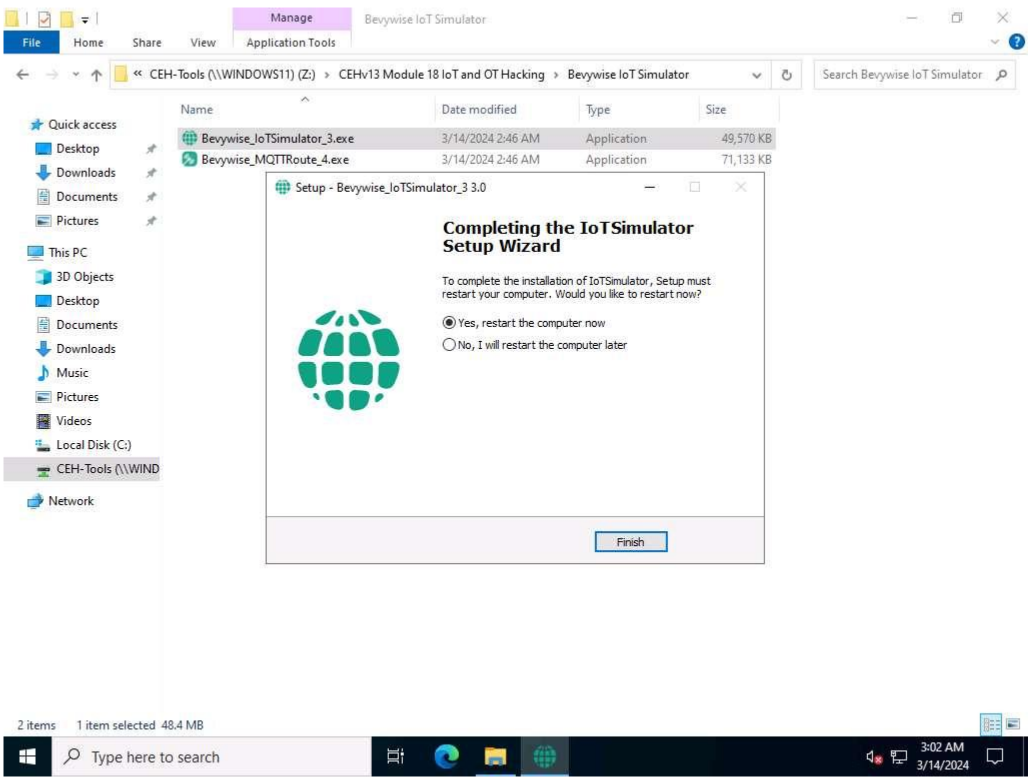
2 items   1 item selected 48.4 MB

11. If **Open File - Security Warning** popup appears, click **Run**..

12. The **Setup-IoTSimulator_3 3.0** setup wizard opens. Select **I accept the agreement** and follow the wizard driven steps.

13. To complete the installation, select **Yes, restart the computer now** and click on **Finish** to complete the installation.

    If restart computer option does not appear, then continue from **Step#16**.


Setup - Bevywise_IoTSimulator_3 3.0

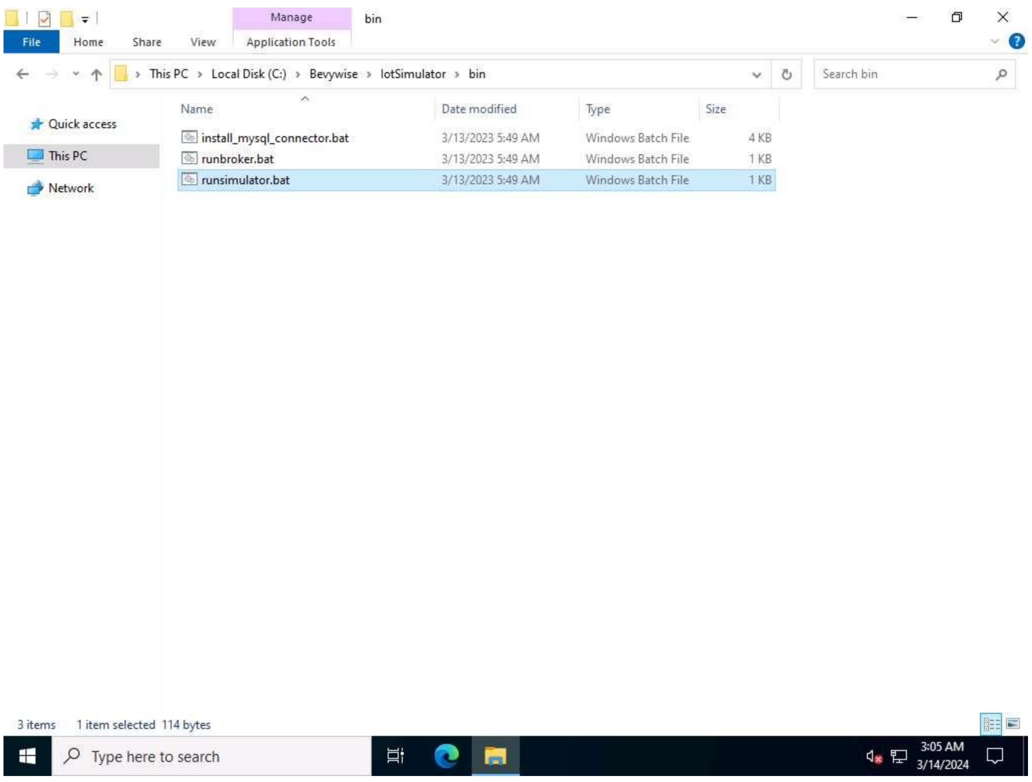
Completing the IoTSimulator Setup Wizard

To complete the installation of IoTSimulator, Setup must restart your computer. Would you like to restart now?

[✓ Yes, restart the computer now]
No, I will restart the computer later

Name	Date modified	Type	Size
Bevywise_IoTSimulator_3.exe	3/14/2024 2:46 AM	Application	49,570 KB
Bevywise_MQTTRoute_4.exe	3/14/2024 2:46 AM	Application	71,133 KB

14. After restarting, 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.


This PC > Local Disk (C:) > Bevywise > IotSimulator > bin

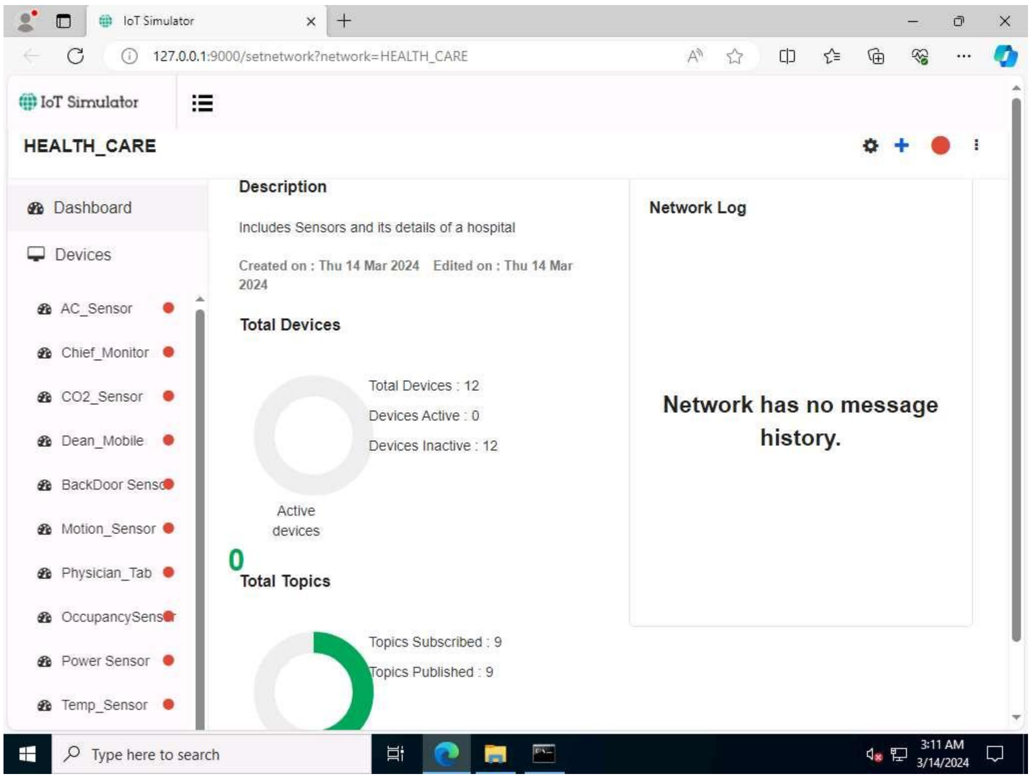
Name	Date modified	Type	Size
install_mysql_connector.bat	3/13/2023 5:49 AM	Windows Batch File	4 KB
runbroker.bat	3/13/2023 5:49 AM	Windows Batch File	1 KB
runsimulator.bat	3/13/2023 5:49 AM	Windows Batch File	1 KB

15. Upon double-clicking 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 **OK** to open the URL **http://127.0.0.1:9000/setnetwork?network=HEALTH_CARE**.

If the URL directly opens in Microsoft Edge browser, then continue.

16. The web interface of the IoT Simulator opens in Edge browser. In the IoT Simulator, you can view the default network named **HEALTH_CARE** and several devices.



IoT Simulator

127.0.0.1:9000/setnetwork?network=HEALTH_CARE

HEALTH_CARE

Dashboard
Devices
AC_Sensor
Chief_Monitor
CO2_Sensor
Dean_Mobile
BackDoor Sensor
Motion_Sensor
Physician_Tab
OccupancySensor
Power Sensor
Temp_Sensor

Description

Includes Sensors and its details of a hospital

Created on : Thu 14 Mar 2024 Edited on : Thu 14 Mar 2024

Total Devices

Total Devices : 12
Devices Active : 0
Devices Inactive : 12

Active devices
0

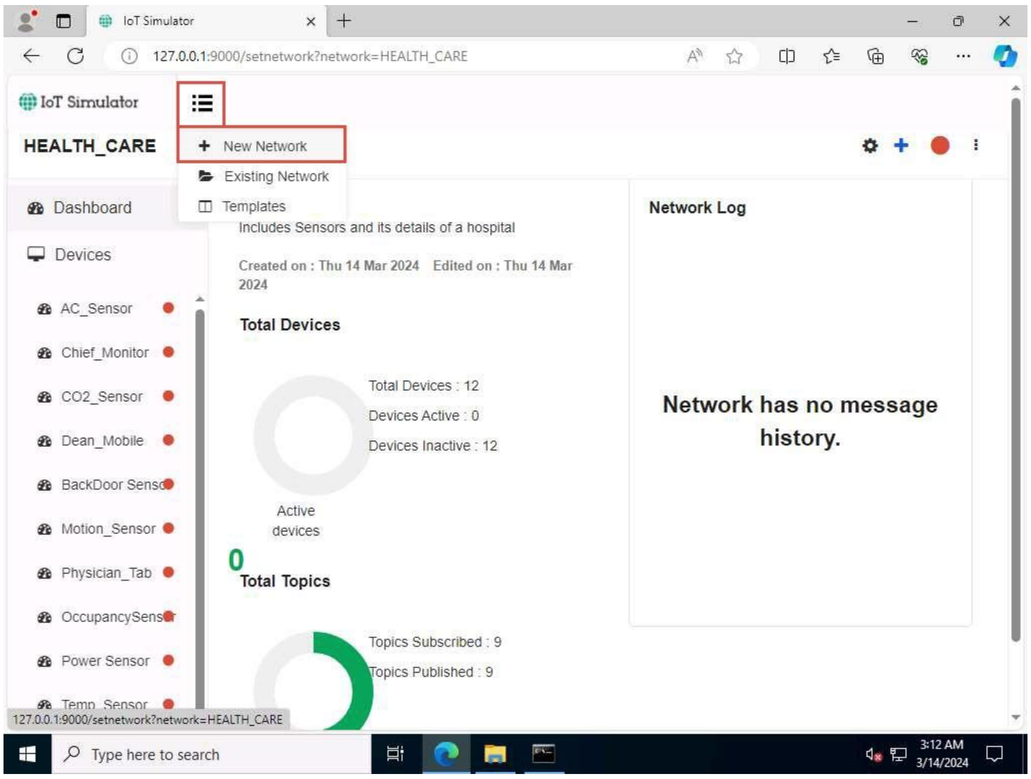
Total Topics

Topics Subscribed : 9
Topics Published : 9

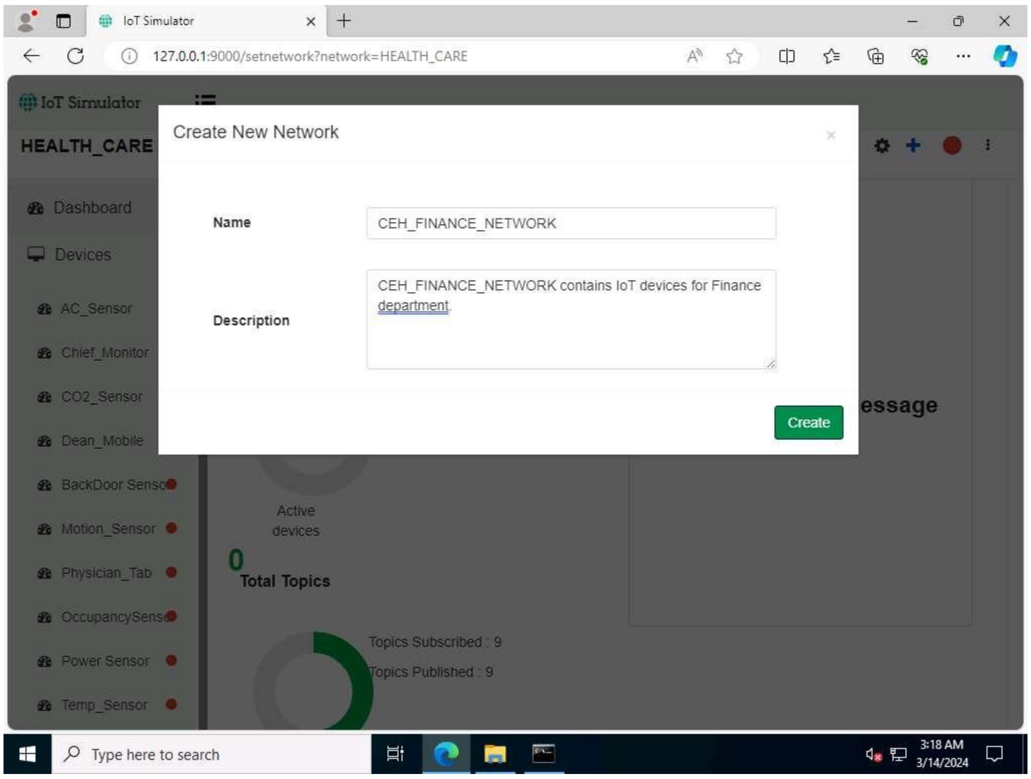
Network Log

Network has no message history.

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



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

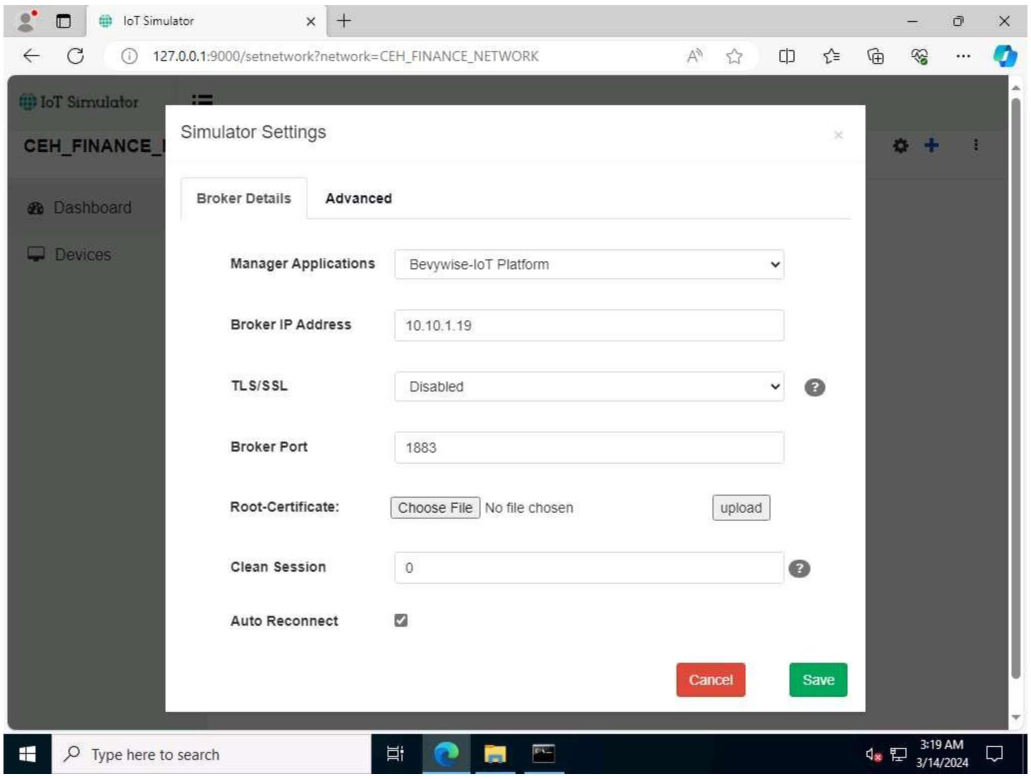


Name: CEH_FINANCE_NETWORK

Description: CEH_FINANCE_NETWORK contains IoT devices for Finance department.

19. In the next screen, we will setup the **Simulator Settings**. Set the **Broker IP Address** as **10.10.1.19** (the IP address of the **Windows Server 2019**). Since we have installed the Broker on the web server, the created network will interact

with the server using MQTT Broker. Do not change default settings and click on **Save**.


IoT Simulator

127.0.0.1:9000/setnetwork?network=CEH_FINANCE_NETWORK

Simulator Settings

Broker Details | Advanced

Manager Applications: Bevywise-IoT Platform

Broker IP Address: 10.10.1.19

TLS/SSL: Disabled

Broker Port: 1883

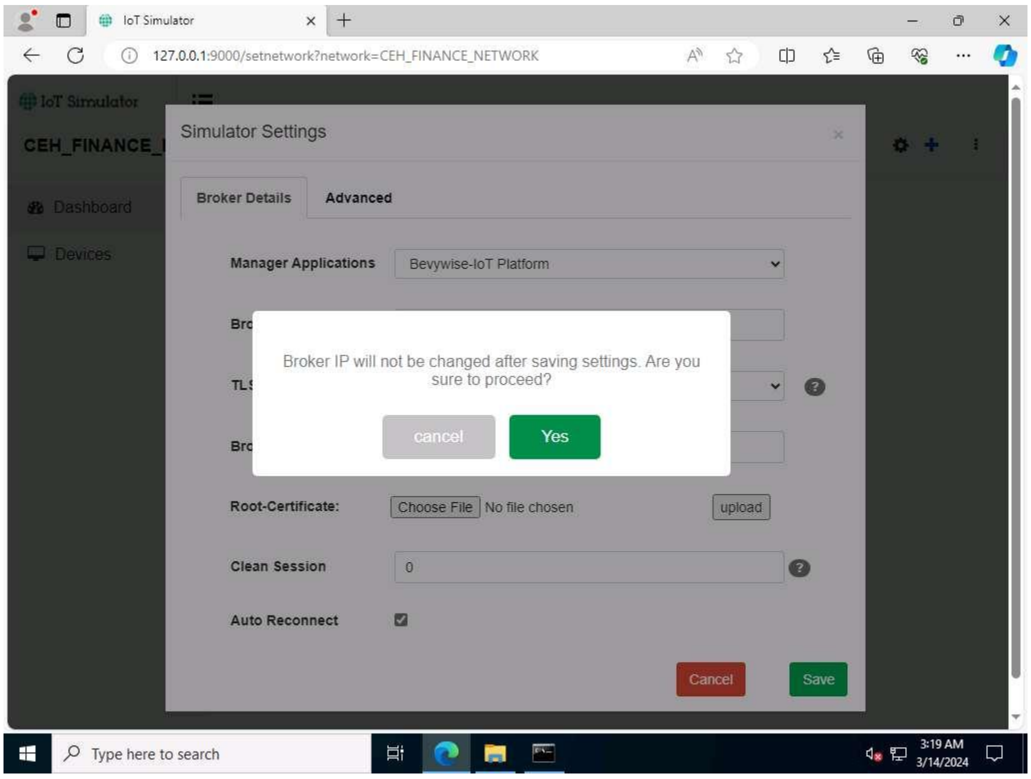
Root-Certificate: Choose File No file chosen upload

Clean Session: 0

Auto Reconnect: ☑

Cancel Save

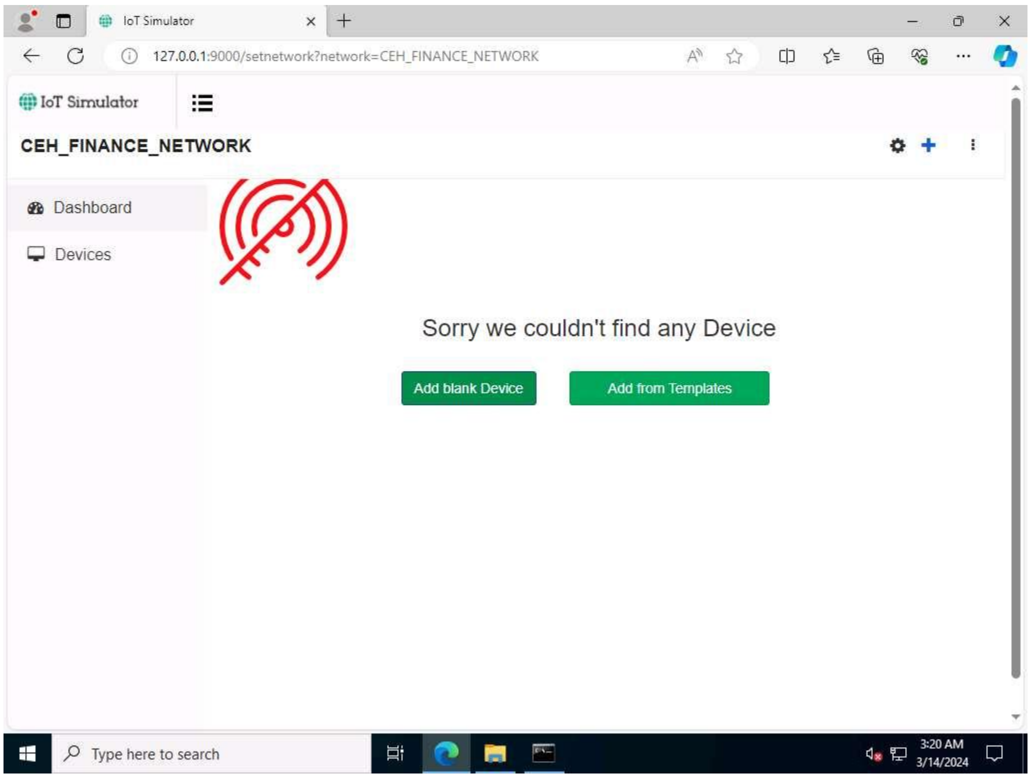
20. To proceed with the network creation, click on **Yes**.


Broker IP will not be changed after saving settings. Are you sure to proceed?

cancel Yes

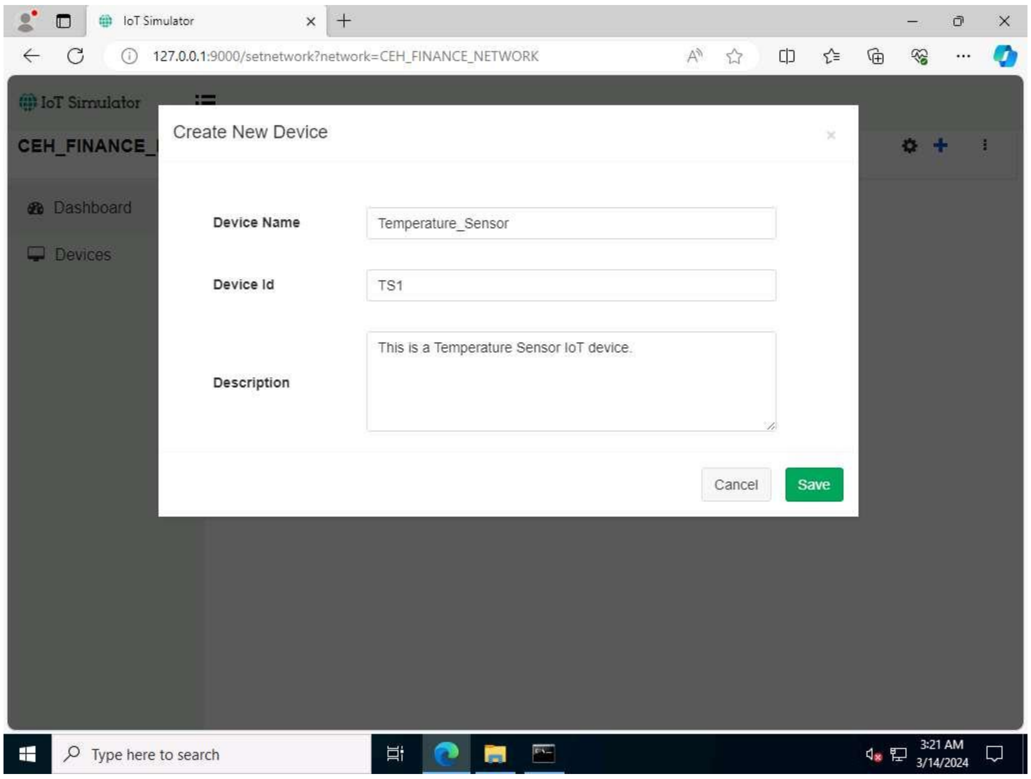
If **Configuration Saved** pop-up appears. Click on **OK** to continue. This step completes the creation of the virtual IoT network.

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



Sorry we couldn't find any Device

22. The **Create New Device** popup opens. Type the device name (here, we use **Temperature_Sensor**), enter Device Id (here, we use **TS1**), provide a **Description** and click on **Save**.

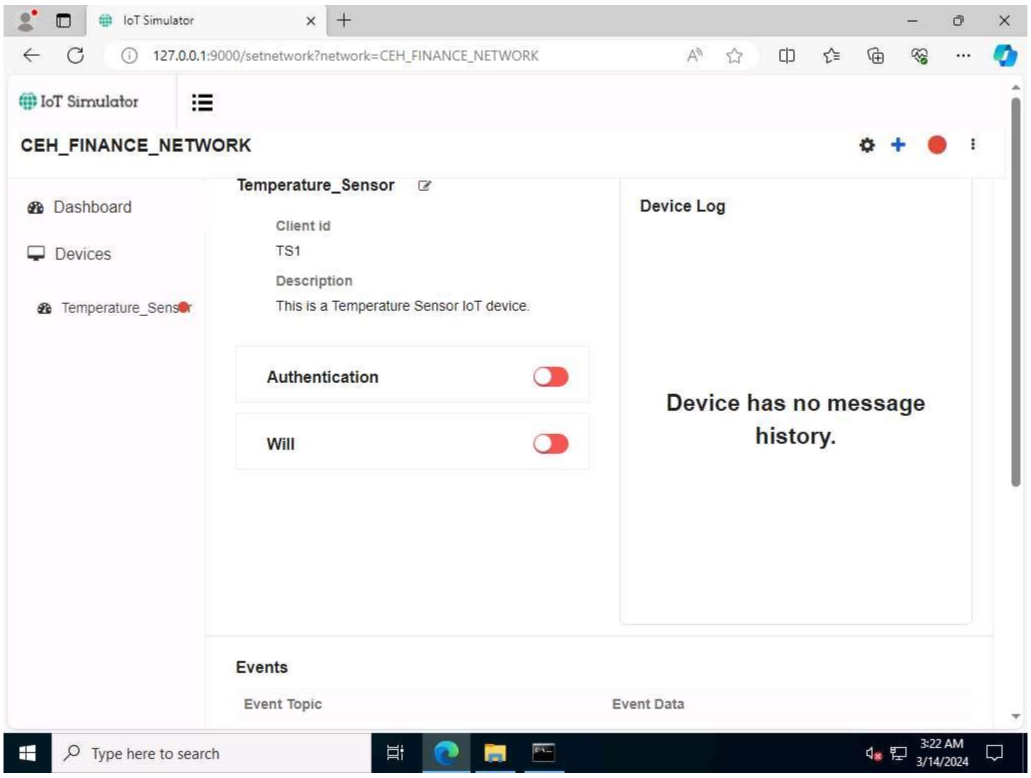

Create New Device

Device Name: Temperature_Sensor

Device Id: TS1

Description: This is a Temperature Sensor IoT device.

23. The device will be added to the **CEH_FINANCE_NETWORK**.


CEH_FINANCE_NETWORK

Temperature_Sensor

Client id
TS1

Description
This is a Temperature Sensor IoT device.

Authentication

Will

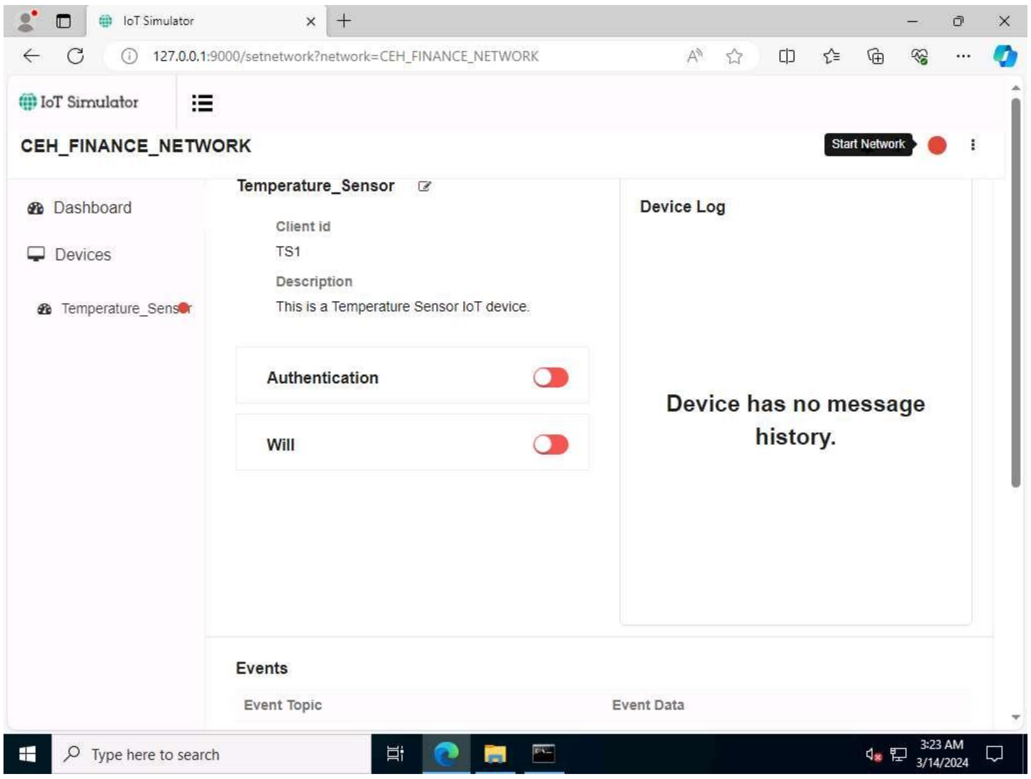
Device Log

Device has no message history.

Events

Event Topic | Event Data

24. 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.



Temperature_Sensor

Client id
TS1

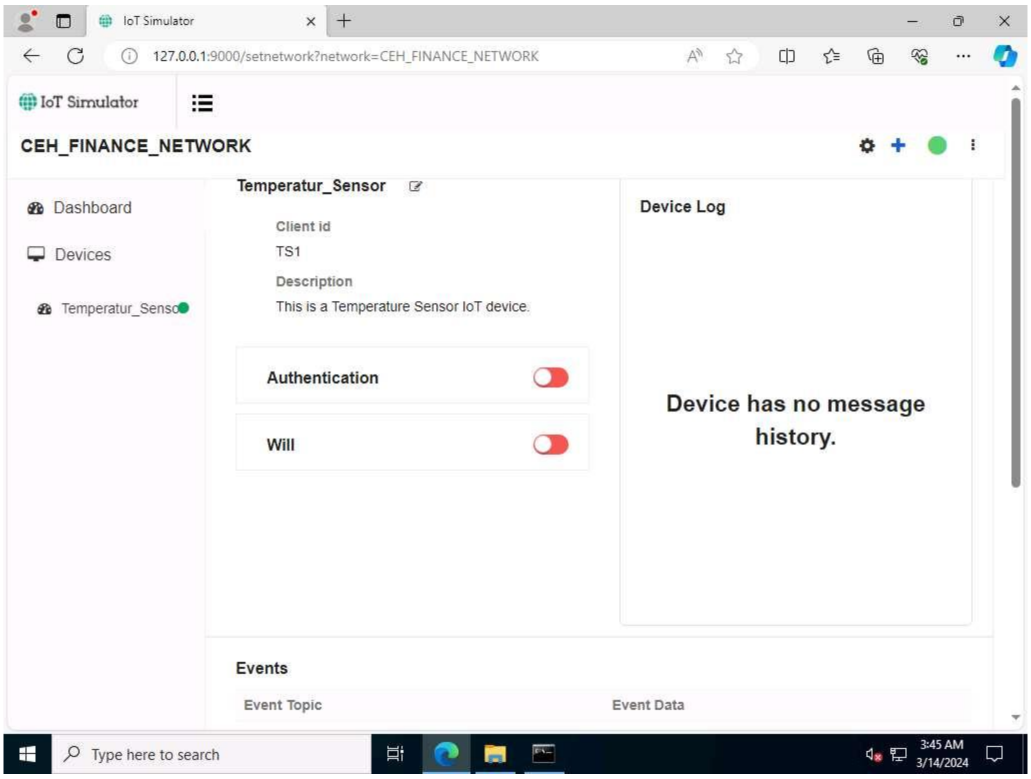
Description
This is a Temperature Sensor IoT device.

Device Log

Device has no message history.

Event Topic | Event Data

25. 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**.



Temperatur_Sensor

Client id
TS1

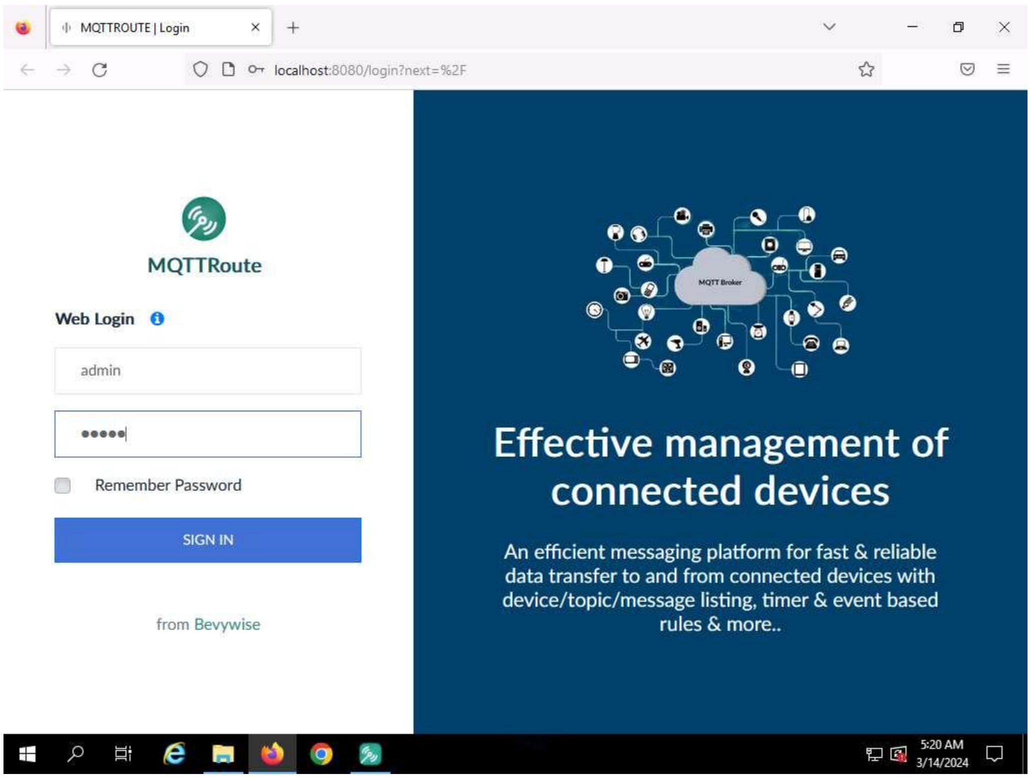
Description
This is a Temperature Sensor IoT device.

Device Log

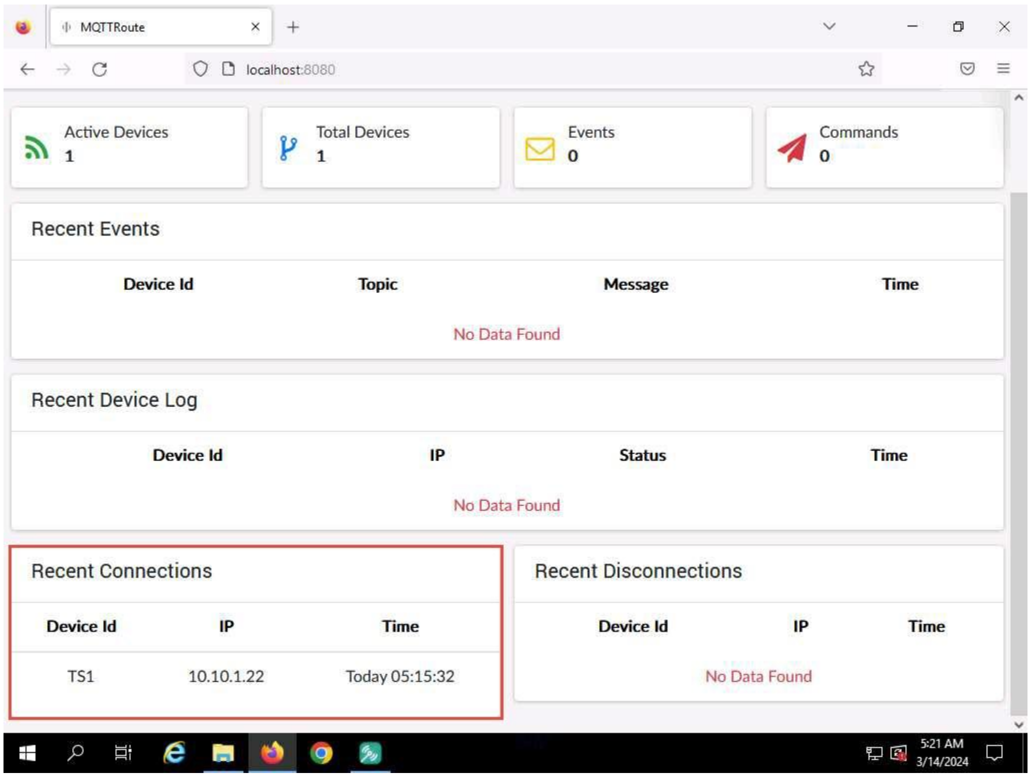
Device has no message history.

Event Topic | Event Data

26. Next, switch to the Windows Server 2019 machine. Open a web browser, and go to **http://localhost:8080** and login using **admin/admin** (here, we are using **Firefox** Browser).


MQTTROUTE | Login
localhost:8080/login?next=%2F
MQTTRoute
Web Login
admin
Remember Password
SIGN IN
from Bevywise
MQTT Broker
Effective management of connected devices
An efficient messaging platform for fast & reliable data transfer to and from connected devices with device/topic/message listing, timer & event based rules & more..
5:20 AM
3/14/2024

27. Since the Broker was **left running**, you can see a connection request from machine **10.10.1.22** for the device **TS1** under **Recent Connections** section.



Recent Events

Device Id	Topic	Message	Time
No Data Found			

Recent Device Log

Device Id	IP	Status	Time
No Data Found			

Recent Connections

Device Id	IP	Time
TS1	10.10.1.22	Today 05:15:32

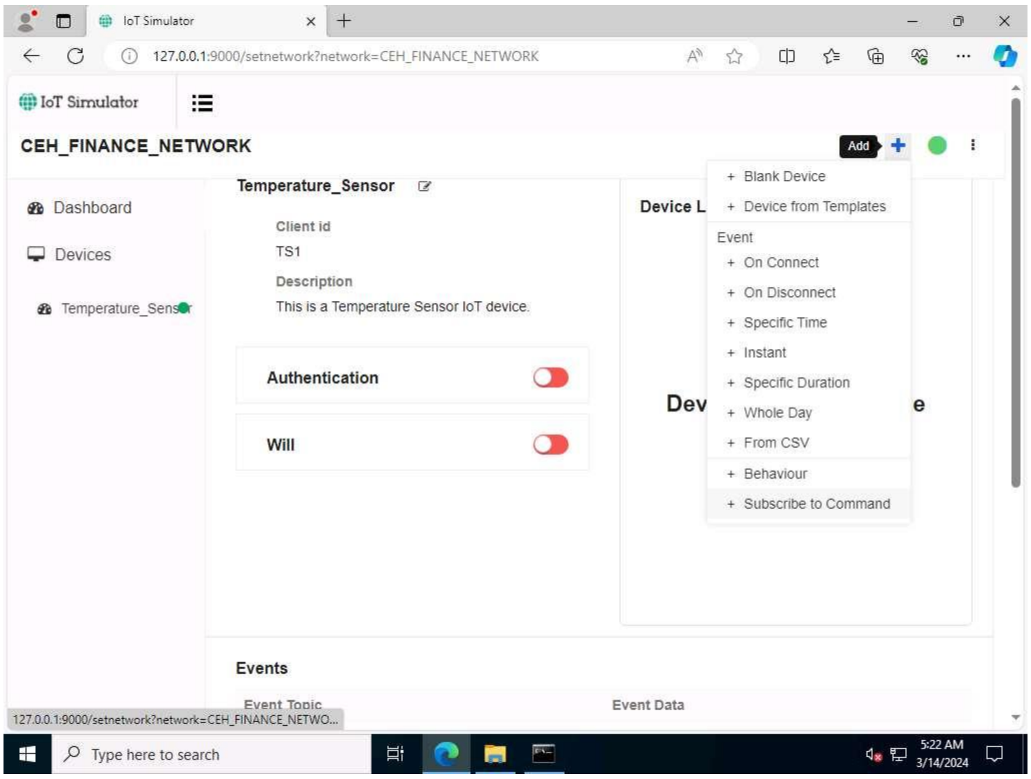
Recent Disconnections

Device Id	IP	Time
No Data Found		

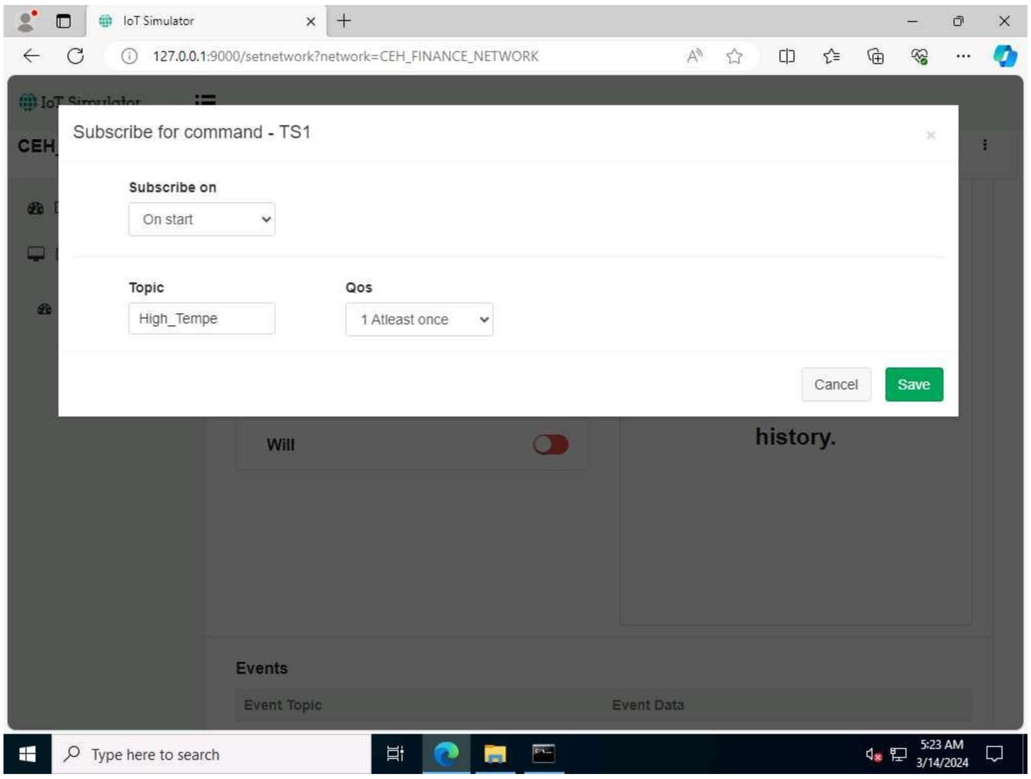
28. Switch back to Windows Server 2022 machine.

29. Next, we will create the **Subscribe command** for the device Temperature_Sensor.

30. Click on the **Plus** icon in **the top right corner** and select the **Subscribe to Command** option.



31. The **Subscribe for command - TS1** popup opens. 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**.

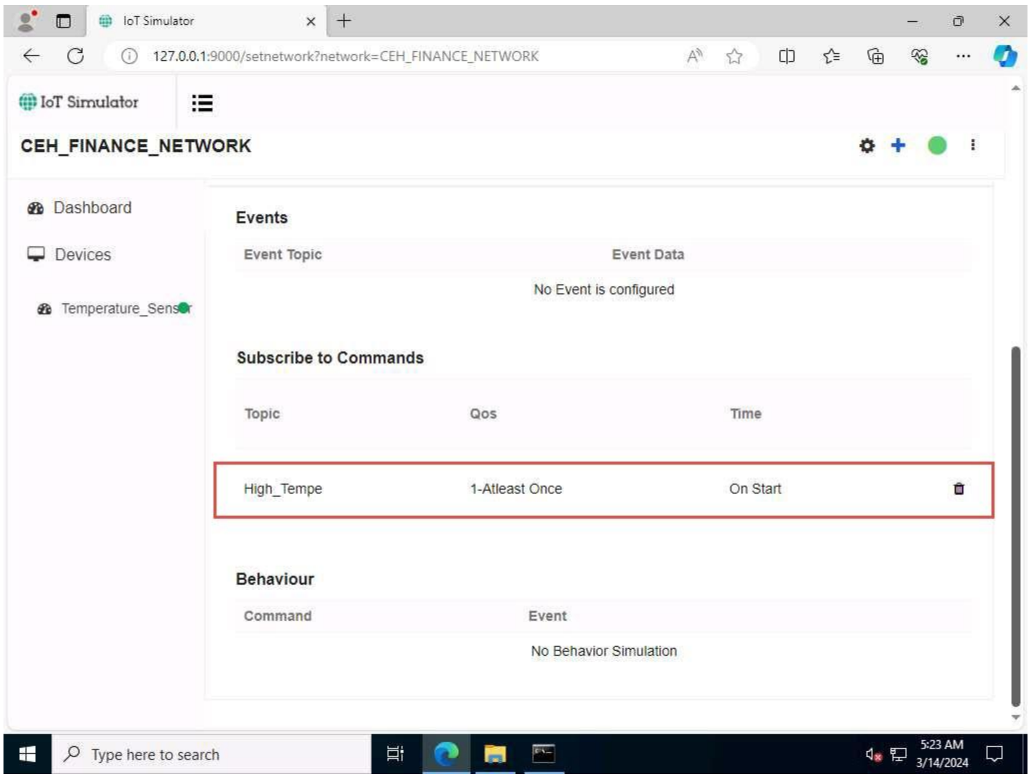

Subscribe for command - TS1

Subscribe on: On start

Topic: High_Tempe

Qos: 1 Atleast once

32. Scroll down the page, you can see the **Topic** added under the **Subscribe to Commands** section.



Events

Event Topic	Event Data
No Event is configured	

Subscribe to Commands

Topic	Qos	Time
High_Tempe	1-Atleast Once	On Start

Behaviour

Command	Event
No Behavior Simulation	

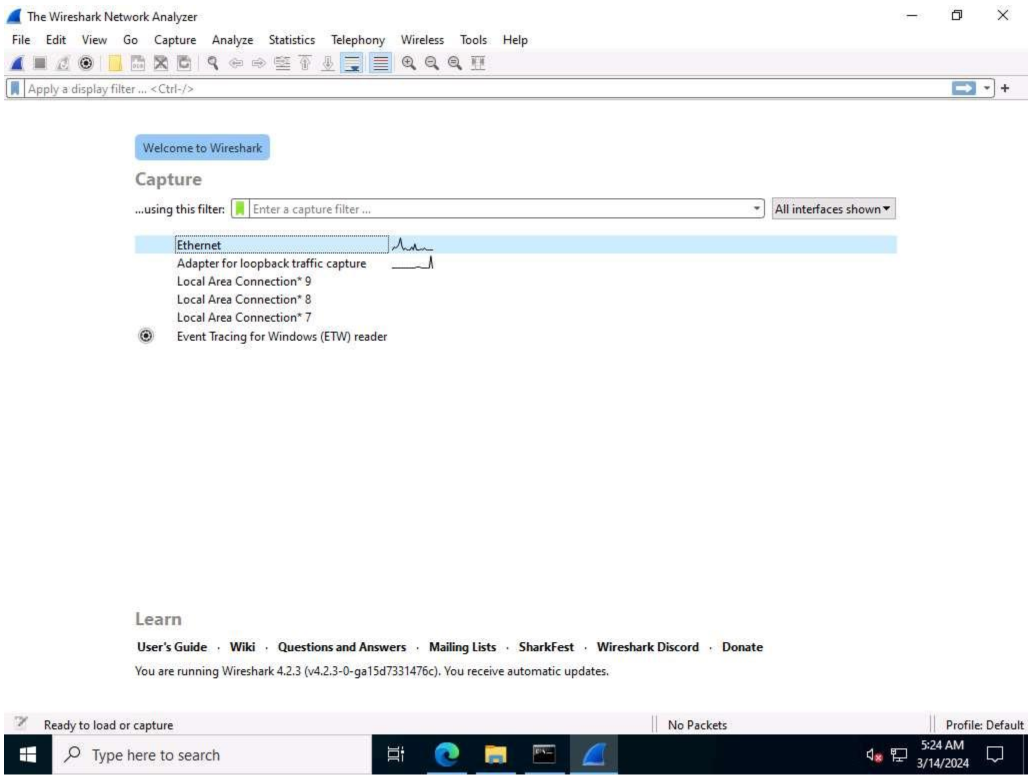
33. Next, we will capture the traffic between the **virtual IoT network and the MQTT Broker** to monitor the secure communication.

34. Minimise the Edge browser. Click **Type here to search** field on the **Desktop**, search for **wireshark** in the search bar and select **Wireshark** from the results.

35. The Wireshark Application window appears, select the **Ethernet** as interface.

    Make sure you have selected interface which has **10.10.1.22** as the IP address.

    If Software update popup appears click on **Skip this version**.

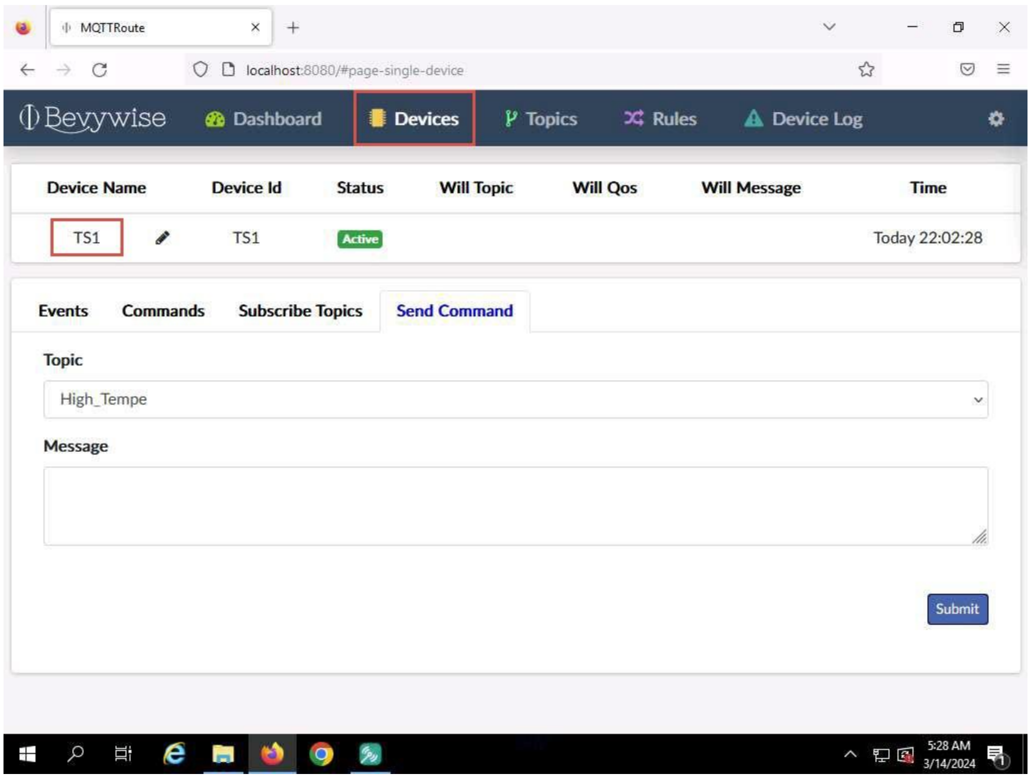

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Ethernet
Adapter for loopback traffic capture
Local Area Connection* 9
Local Area Connection* 8
Local Area Connection* 7
Event Tracing for Windows (ETW) reader
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate
Ready to load or capture
No Packets
Profile: Default
Type here to search
5:24 AM
3/14/2024

You are running Wireshark 4.2.3 (v4.2.3-0-ga15d7331476c). You receive automatic updates.

36. Click on the **Start Wireshark** icon to start the capturing packets, leave the Wireshark running.

37. Leave the IoT simulator running and switch to the Windows Server 2019 machine.

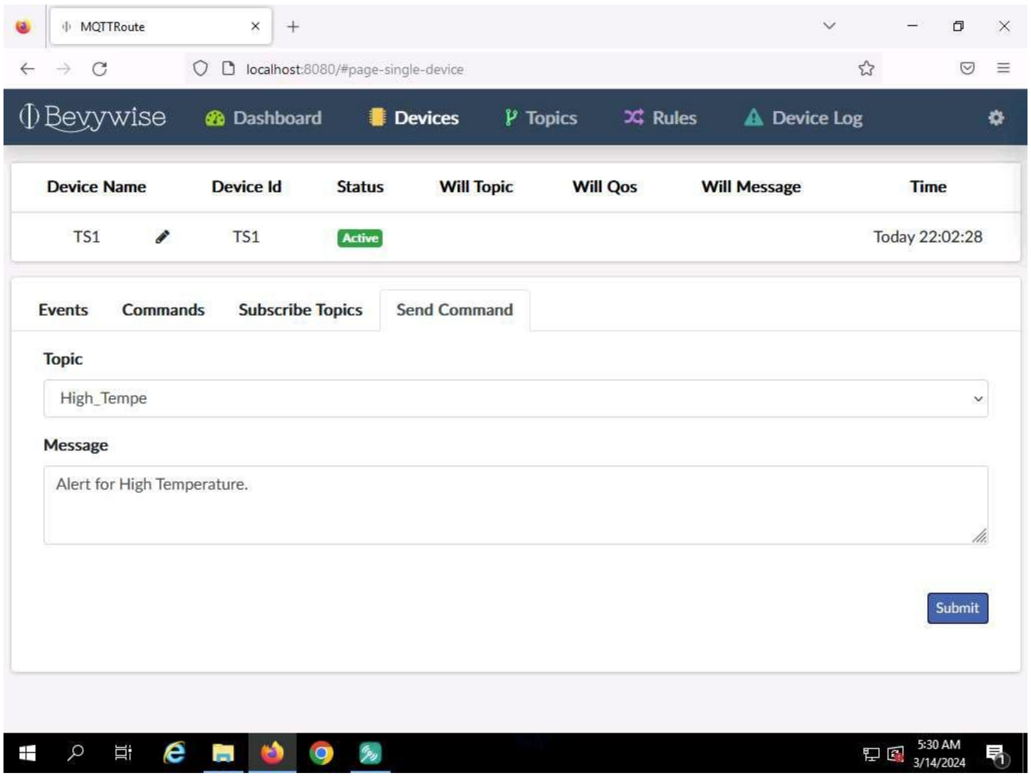
38. Navigate to **Devices** menu and click on connected device i.e.**TS1**.



Device Name	Device Id	Status	Will Topic	Will Qos	Will Message	Time
TS1	TS1	Active				Today 22:02:28

39. Now, we will send the command to **TS1** using the **High_Tempe** topic.

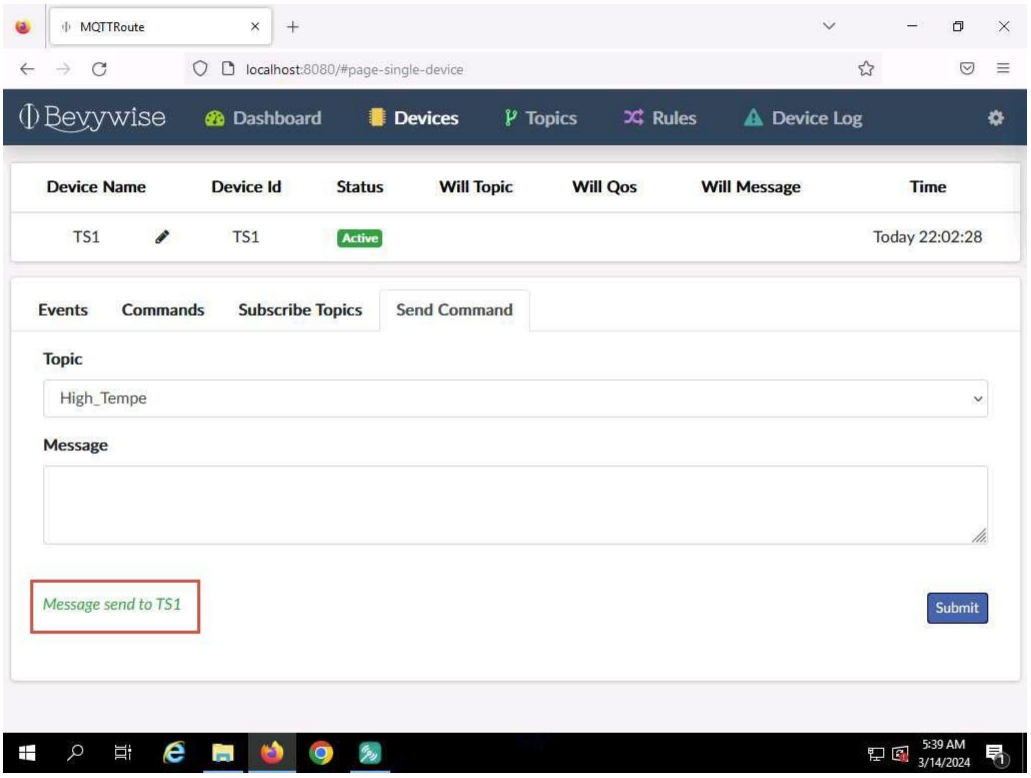
40. In **Send Command** section, select **Topic** as **High_Tempe**, type **Alert for High Temperature** in **Message** field and click on the **Submit** button.



Device Name	Device Id	Status	Will Topic	Will Qos	Will Message	Time
TS1	TS1	Active				Today 22:02:28

Alert for High Temperature.

41. **Message sent to TS1** appears under **Message** box which indicates that the message was successfully sent to TS1.



MQTTRoute

localhost:8080/#page-single-device

Device Name	Device Id	Status	Will Topic	Will Qos	Will Message	Time
TS1	TS1	Active				Today 22:02:28

Topic: High_Tempe

Message:

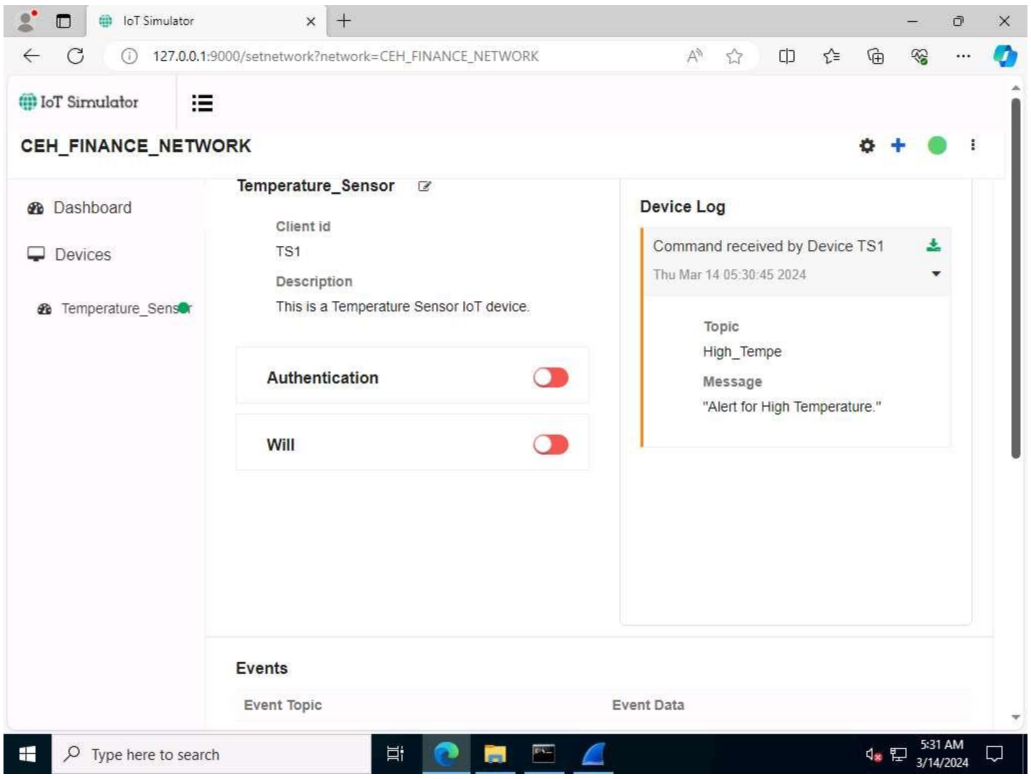
Message send to TS1

42. The message has been sent to the device using this topic.

43. Next, switch to Windows Server 2022 machine.

44. We have left the IoT simulator running in the web browser. To see the alert message, maximise the Edge browser and expand the arrow under the connected **Temperature_Sensor**, **Device Log** section.

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



127.0.0.1:9000/setnetwork?network=CEH_FINANCE_NETWORK

CEH_FINANCE_NETWORK

Temperature_Sensor

Client id
TS1

Description
This is a Temperature Sensor IoT device.

Authentication

Will

Device Log

Command received by Device TS1
Thu Mar 14 05:30:45 2024

Topic
High_Tempe

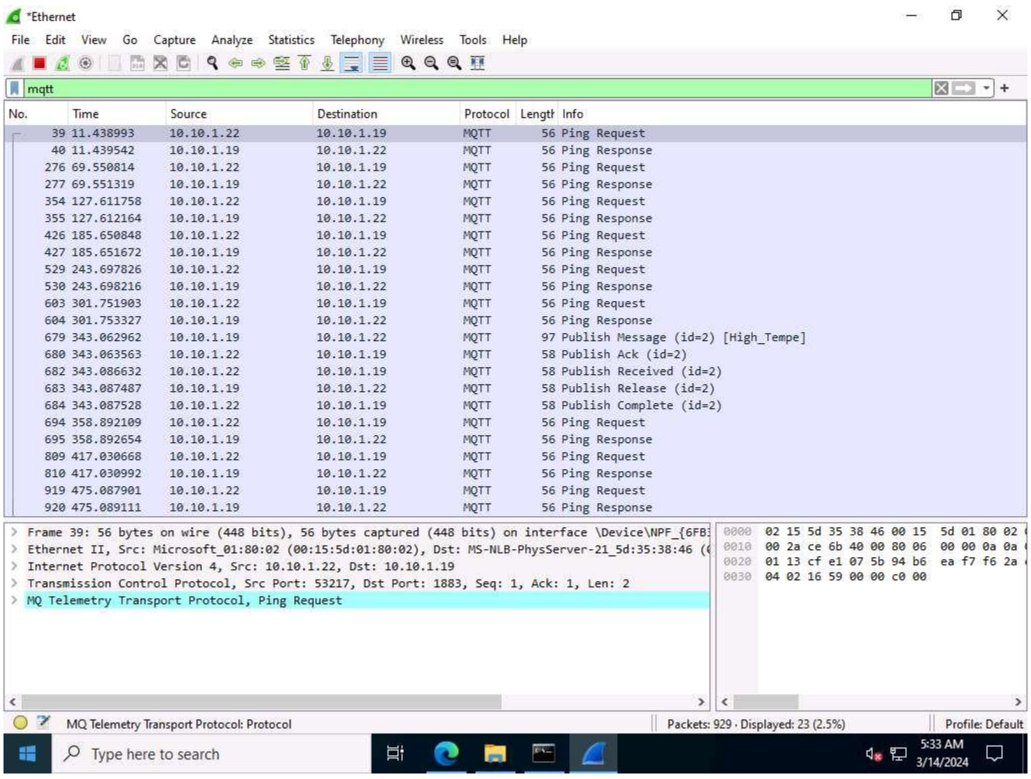
Message
"Alert for High Temperature."

Events

Event Topic	Event Data

46. To verify the communication, we have executed **Wireshark** application, switch to the Wireshark traffic capturing window.

47. Type **mqtt** under the **filter** field and press **Enter**. To display only the MQTT protocol packets.



Filter: mqtt

```
No.   Time        Source       Destination  Protocol Length Info
   39 11.438993   10.10.1.22   10.10.1.19   MQTT       56 Ping Request
   40 11.439542   10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  276 69.550814   10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  277 69.551319   10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  354 127.611758  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  355 127.612164  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  426 185.650848  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  427 185.651672  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  529 243.697826  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  530 243.698216  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  603 301.751903  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  604 301.753327  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  679 343.062962  10.10.1.19   10.10.1.22   MQTT       97 Publish Message (id=2) [High_Tempe]
  680 343.063563  10.10.1.22   10.10.1.19   MQTT       58 Publish Ack (id=2)
  682 343.086632  10.10.1.22   10.10.1.19   MQTT       58 Publish Received (id=2)
  683 343.087487  10.10.1.19   10.10.1.22   MQTT       58 Publish Release (id=2)
  684 343.087528  10.10.1.22   10.10.1.19   MQTT       58 Publish Complete (id=2)
  694 358.892109  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  695 358.892654  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  809 417.030668  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  810 417.030992  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  919 475.087901  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  920 475.089111  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
```

```
> Frame 39: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{6FB
> Ethernet II, Src: Microsoft_01:80:02 (00:15:5d:01:80:02), Dst: MS-NLB-PhysServer-21_5d:35:38:46 (
> Internet Protocol Version 4, Src: 10.10.1.22, Dst: 10.10.1.19
> Transmission Control Protocol, Src Port: 53217, Dst Port: 1883, Seq: 1, Ack: 1, Len: 2
> MQ Telemetry Transport Protocol, Ping Request
```

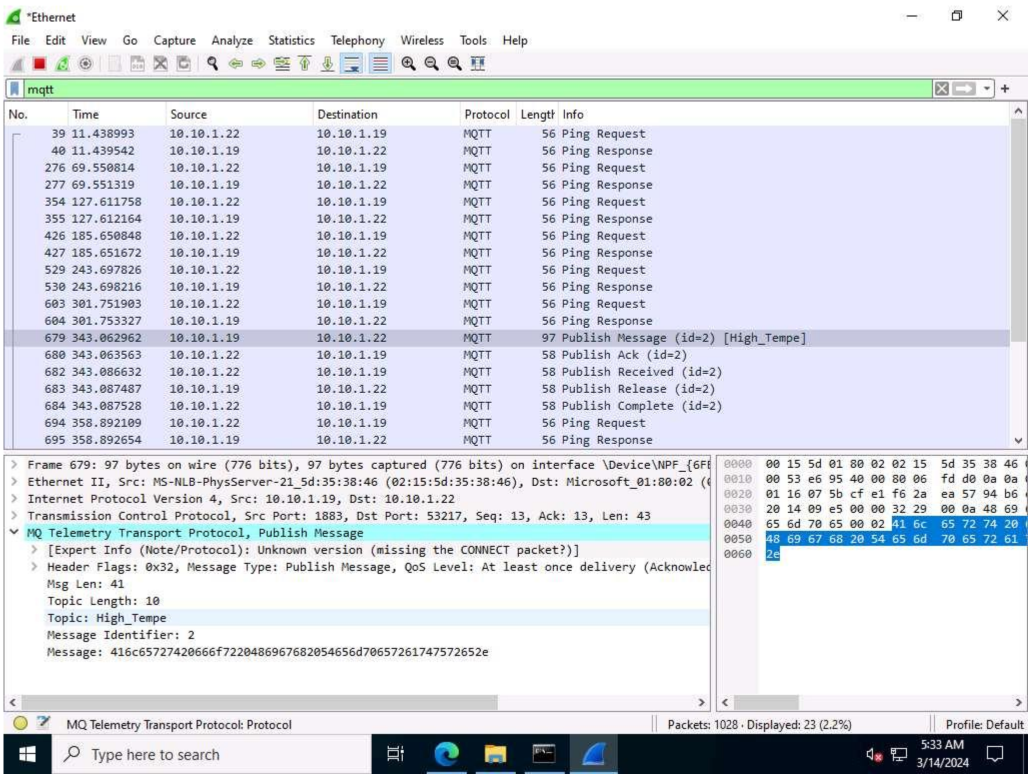
```
0000  02 15 5d 35 38 46 00 15  5d 01 80 02
0010  00 2a ce 6b 40 00 80 06  00 00 0a 0a
0020  01 13 cf e1 07 5b 94 b6  ea f7 f6 2a
0030  04 02 16 59 00 00 c0 00
```

MQ Telemetry Transport Protocol: Protocol | Packets: 929 · Displayed: 23 (2.5%) | Profile: Default

48. Select any **Publish Message** packet from the **Packet List** pane. In the **Packet Details** pane at the middle of the window, expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.

49. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Topic Length**, **Topic**, and **Message**.

50. Publish Message can be used to obtain the message sent by the MQTT client to the broker.



*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
mqtt

```
No.  Time        Source      Destination  Protocol Length Info
 39 11.438993   10.10.1.22  10.10.1.19   MQTT     56 Ping Request
 40 11.439542   10.10.1.19  10.10.1.22   MQTT     56 Ping Response
276 69.550814   10.10.1.22  10.10.1.19   MQTT     56 Ping Request
277 69.551319   10.10.1.19  10.10.1.22   MQTT     56 Ping Response
354 127.611758  10.10.1.22  10.10.1.19   MQTT     56 Ping Request
355 127.612164  10.10.1.19  10.10.1.22   MQTT     56 Ping Response
426 185.650848  10.10.1.22  10.10.1.19   MQTT     56 Ping Request
427 185.651672  10.10.1.19  10.10.1.22   MQTT     56 Ping Response
529 243.697826  10.10.1.22  10.10.1.19   MQTT     56 Ping Request
530 243.698216  10.10.1.19  10.10.1.22   MQTT     56 Ping Response
603 301.751903  10.10.1.22  10.10.1.19   MQTT     56 Ping Request
604 301.753327  10.10.1.19  10.10.1.22   MQTT     56 Ping Response
679 343.062962  10.10.1.19  10.10.1.22   MQTT     97 Publish Message (id=2) [High_Tempe]
680 343.063563  10.10.1.22  10.10.1.19   MQTT     58 Publish Ack (id=2)
682 343.086632  10.10.1.22  10.10.1.19   MQTT     58 Publish Received (id=2)
683 343.087487  10.10.1.19  10.10.1.22   MQTT     58 Publish Release (id=2)
684 343.087528  10.10.1.22  10.10.1.19   MQTT     58 Publish Complete (id=2)
694 358.892109  10.10.1.22  10.10.1.19   MQTT     56 Ping Request
695 358.892654  10.10.1.19  10.10.1.22   MQTT     56 Ping Response
```

```
> Frame 679: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface \Device\NPF_{6F
> Ethernet II, Src: MS-NLB-PhysServer-21_5d:35:38:46 (02:15:5d:35:38:46), Dst: Microsoft_01:80:02 (
> Internet Protocol Version 4, Src: 10.10.1.19, Dst: 10.10.1.22
> Transmission Control Protocol, Src Port: 1883, Dst Port: 53217, Seq: 13, Ack: 13, Len: 43
v MQ Telemetry Transport Protocol, Publish Message
  > [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]
  > Header Flags: 0x32, Message Type: Publish Message, QoS Level: At least once delivery (Acknowled
    Msg Len: 41
    Topic Length: 10
    Topic: High_Tempe
    Message Identifier: 2
    Message: 416c65727420666f72204869676820546656d70657261747572652e
```

```
0000  00 15 5d 01 80 02 02 15  5d 35 38 46
0010  00 53 e6 95 40 00 80 06  fd d0 0a 0a
0020  01 16 07 5b cf e1 f6 2a  ea 57 94 b6
0030  20 14 09 e5 00 00 32 29  00 0a 48 69
0040  65 6d 70 65 00 02 41 6c  65 72 74 20
0050  48 69 67 68 20 54 65 6d  70 65 72 61
0060  2e
```

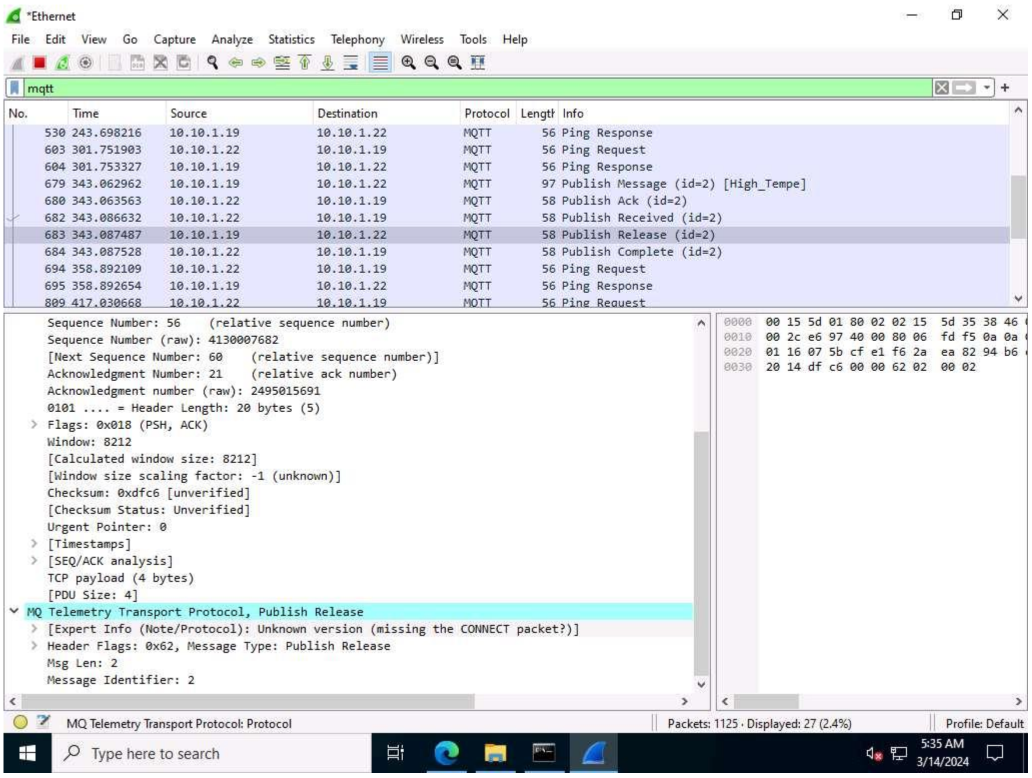
MQ Telemetry Transport Protocol: Protocol
Packets: 1028 · Displayed: 23 (2.2%)
Profile: Default
Type here to search
5:33 AM
3/14/2024

Note: After establishing a successful connection with the MQTT broker, the MQTT client can publish messages. The headers in the Publish Message packet are given below:

- Header Flags: Contains information regarding the MQTT control packet type.
- DUP flag: If the DUP flag is 0, it indicates the first attempt at sending this PUBLISH packet; if the flag is 1, it indicates a possible re-attempt at sending the message.
- QoS: Determines the assurance level of a message.
- Retain Flag: If the retain flag is set to 1, the server must store the message and its QoS, so it can cater to future subscriptions matching the topic.
- Topic Name: Contains a UTF-8 string that can also include forward slashes when it needs to be hierarchically structured.
- Message: Contains the actual data to be transmitted.
- Payload: Contains the message that is being published.

51. Select any **Publish Release** packet from the **Packet List** pane. In the **Packet Details** pane at the middle of the window, expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.

52. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Message Type**, **Message Identifier**.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
mqtt
Type here to search

```
No.   Time          Source        Destination   Protocol  Length Info
  530 243.698216    10.10.1.19    10.10.1.22    MQTT      56 Ping Response
  603 301.751903    10.10.1.22    10.10.1.19    MQTT      56 Ping Request
  604 301.753327    10.10.1.19    10.10.1.22    MQTT      56 Ping Response
  679 343.062962    10.10.1.19    10.10.1.22    MQTT      97 Publish Message (id=2) [High_Tempe]
  680 343.063563    10.10.1.22    10.10.1.19    MQTT      58 Publish Ack (id=2)
  682 343.086632    10.10.1.22    10.10.1.19    MQTT      58 Publish Received (id=2)
  683 343.087487    10.10.1.19    10.10.1.22    MQTT      58 Publish Release (id=2)
  684 343.087528    10.10.1.22    10.10.1.19    MQTT      58 Publish Complete (id=2)
  694 358.892109    10.10.1.22    10.10.1.19    MQTT      56 Ping Request
  695 358.892654    10.10.1.19    10.10.1.22    MQTT      56 Ping Response
  809 417.030668    10.10.1.22    10.10.1.19    MQTT      56 Ping Request
```

```
    Sequence Number: 56    (relative sequence number)
    Sequence Number (raw): 4130007682
    [Next Sequence Number: 60    (relative sequence number)]
    Acknowledgment Number: 21    (relative ack number)
    Acknowledgment number (raw): 2495015691
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 8212
    [Calculated window size: 8212]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xdfc6 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (4 bytes)
    [PDU Size: 4]
v MQ Telemetry Transport Protocol, Publish Release
  > [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]
  > Header Flags: 0x62, Message Type: Publish Release
    Msg Len: 2
    Message Identifier: 2
```

```
0000  00 15 5d 01 80 02 02 15  5d 35 38 46
0010  00 2c e6 97 40 00 80 06  fd f5 0a 0a
0020  01 16 07 5b cf e1 f6 2a  ea 82 94 b6
0030  20 14 df c6 00 00 62 02  00 02
```

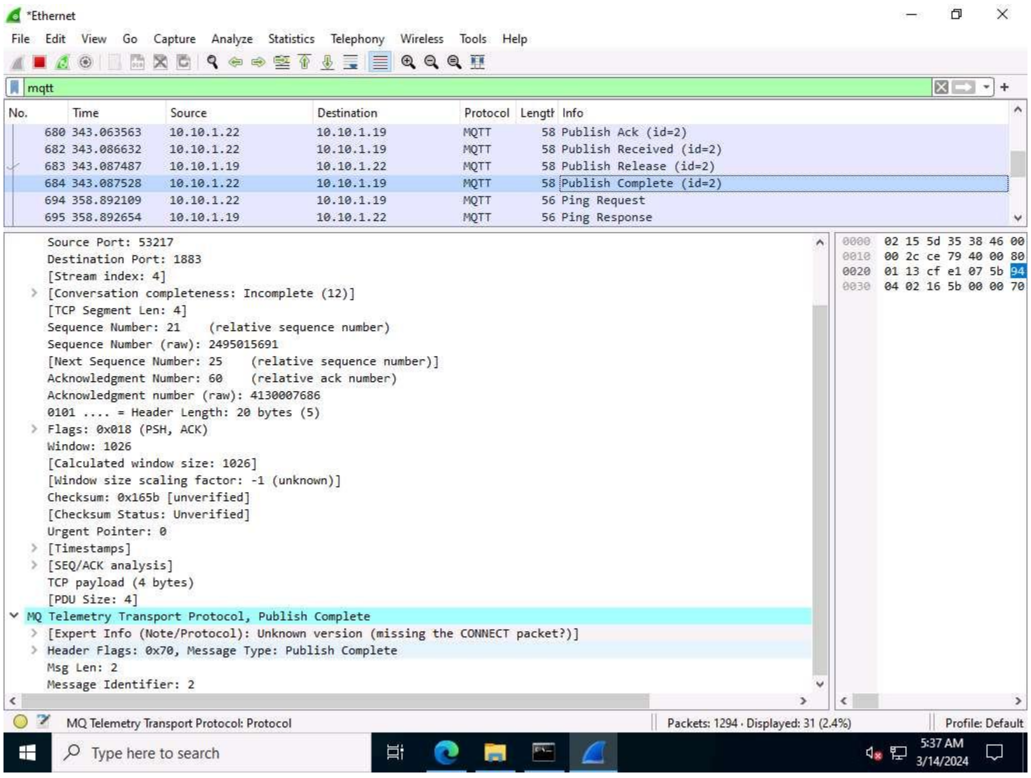
MQ Telemetry Transport Protocol: Protocol | Packets: 1125 · Displayed: 27 (2.4%) | Profile: Default

5:35 AM
3/14/2024

Note: A Publish Release (PUBREL) packet is the response to a Publish Received (PUBREC) packet.

53. Now, scroll down, look for the **Publish Complete** packet from the **Packet List** pane, and click on it. In the **Packet Details** pane at the middle of the window, expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.

54. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len** and **Message Identifier**.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
mqtt

```
No.  Time        Source      Destination  Protocol Length Info
680 343.063563  10.10.1.22   10.10.1.19   MQTT     58 Publish Ack (id=2)
682 343.086632  10.10.1.22   10.10.1.19   MQTT     58 Publish Received (id=2)
683 343.087487  10.10.1.19   10.10.1.22   MQTT     58 Publish Release (id=2)
684 343.087528  10.10.1.22   10.10.1.19   MQTT     58 Publish Complete (id=2)
694 358.892109  10.10.1.22   10.10.1.19   MQTT     56 Ping Request
695 358.892654  10.10.1.19   10.10.1.22   MQTT     56 Ping Response
```

```
    Source Port: 53217
    Destination Port: 1883
    [Stream index: 4]
  > [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 4]
    Sequence Number: 21    (relative sequence number)
    Sequence Number (raw): 2495015691
    [Next Sequence Number: 25    (relative sequence number)]
    Acknowledgment Number: 60    (relative ack number)
    Acknowledgment number (raw): 4130007686
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 1026
    [Calculated window size: 1026]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x165b [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (4 bytes)
    [PDU Size: 4]
v MQ Telemetry Transport Protocol, Publish Complete
  > [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]
  > Header Flags: 0x70, Message Type: Publish Complete
    Msg Len: 2
    Message Identifier: 2
```

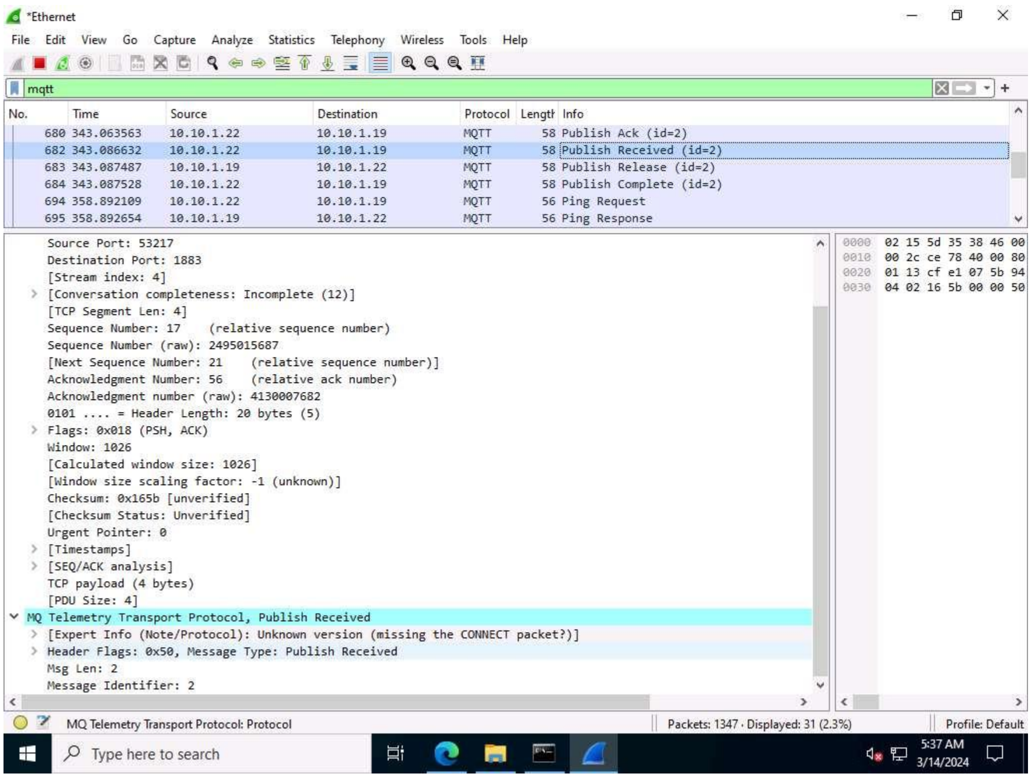
```
0000  02 15 5d 35 38 46 00
0010  00 2c ce 79 40 00 80
0020  01 13 cf e1 07 5b 94
0030  04 02 16 5b 00 00 70
```

MQ Telemetry Transport Protocol: Protocol
Packets: 1294 · Displayed: 31 (2.4%)
Profile: Default
Type here to search
5:37 AM
3/14/2024

Note: The Publish Complete (PUBCOMP) packet is the response to a Publish Release (PUBREL) packet.

55. Now, scroll down, look for the **Publish Received** packet from the **Packet List** pane, and click on it. In the **Packet Details** pane at the middle of the window, expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.

56. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Message Type**, **Msg Len** and **Message Identifier**.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
mqtt

```
No.   Time        Source      Destination  Protocol Length Info
  680 343.063563  10.10.1.22  10.10.1.19   MQTT      58 Publish Ack (id=2)
  682 343.086632  10.10.1.22  10.10.1.19   MQTT      58 Publish Received (id=2)
  683 343.087487  10.10.1.19  10.10.1.22   MQTT      58 Publish Release (id=2)
  684 343.087528  10.10.1.22  10.10.1.19   MQTT      58 Publish Complete (id=2)
  694 358.892109  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  695 358.892654  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
```

```
    Source Port: 53217
    Destination Port: 1883
    [Stream index: 4]
  > [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 4]
    Sequence Number: 17    (relative sequence number)
    Sequence Number (raw): 2495015687
    [Next Sequence Number: 21    (relative sequence number)]
    Acknowledgment Number: 56    (relative ack number)
    Acknowledgment number (raw): 4130007682
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 1026
    [Calculated window size: 1026]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x165b [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (4 bytes)
    [PDU Size: 4]
v MQ Telemetry Transport Protocol, Publish Received
  > [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]
  > Header Flags: 0x50, Message Type: Publish Received
    Msg Len: 2
    Message Identifier: 2
```

```
0000  02 15 5d 35 38 46 00
0010  00 2c ce 78 40 00 80
0020  01 13 cf e1 07 5b 94
0030  04 02 16 5b 00 00 50
```

MQ Telemetry Transport Protocol: Protocol
Packets: 1347 · Displayed: 31 (2.3%)
Profile: Default
Type here to search
5:37 AM
3/14/2024

57. Similarly you can select **Ping Request**, **Ping Response** and **Publish Ack** packets and observe the details.

58. This concludes the demonstration of capturing and analyzing MQTT protocol packets. Here, we analyzed different processes involved in the communication between an MQTT client and an MQTT broker using Wireshark. Understanding these metrics as well as the workflow can help you in quickly identifying the MQTT-related issues.

59. Close all open windows and document all the acquired information.

Question 18.2.1.1

Use Wireshark and Bevywise MQTT Route and Bevywise IoT Simulator to capture and analyze traffic between IoT devices. What is the default TCP port used by Bevywise MQTT Route to establish connection with Bevywise IoT Simulator?

Question 18.2.1.2

Use Wireshark and Bevywise MQTT Route and Bevywise IoT Simulator to capture and analyze traffic between IoT devices. What is the default WebSocket port used by Bevywise MQTT IoT Simulator to establish connection with Bevywise MQTT Route?

Lab 3: Perform IoT Attacks

Lab Scenario

As an ethical hacker or penetration tester, you must have sound knowledge in implementing various techniques to exploit vulnerabilities and launch attacks on target IoT devices or networks.

Potential vulnerabilities in the IoT system can result in major problems for organizations. Most IoT devices come with security issues such as the absence of a proper authentication mechanism or the use of default credentials, absence of a lock-out mechanism, absence of a strong encryption scheme, absence of proper key management systems, and improper physical security.

Lab Objectives

- Perform replay attack on CAN protocol

Overview of IoT Attacks

Owing to the significant growth of the paradigm of the IoT, an increasing number of devices are entering our lives every day. From the automation of homes to healthcare applications, the IoT is everywhere. However, despite the ability of IoT devices to make our lives easier and more comfortable, we cannot underestimate the risk of cyber-attacks. IoT devices lack basic security, thus making them prone to various types of cyber-attacks.

Task 1: Perform Replay Attack on CAN Protocol

The Controller Area Network (CAN) protocol is a robust communication system that allows microcontrollers and devices to interact without a central computer. It uses a message-based approach for reliable data exchange, even in noisy environments. CAN is widely used in automotive industry due to its reliability and simplicity. In modern vehicles, CAN protocol is central to system communication, enabling connections between engine controls, brakes, and infotainment units. However, this interconnectivity can be exploited by hackers to manipulate vehicle functions, posing safety risks.

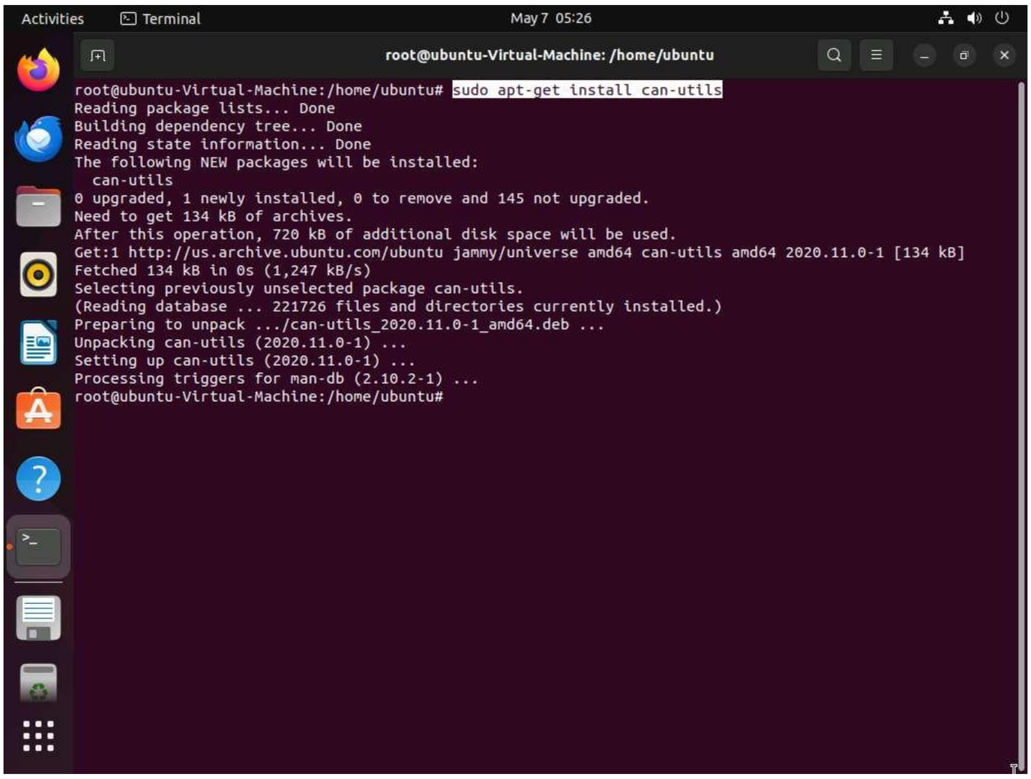
Here, we are using the ICSim tool to simulate CAN protocol and demonstrate how attackers sniff the transmitted packets and perform replay attack to gain basic control over the target.

1. Click Ubuntu to switch to the **Ubuntu** machine and login with **Ubuntu/toor**.

2. In the **Ubuntu** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

3. Run **sudo apt-get install can-utils** to install CAN utility

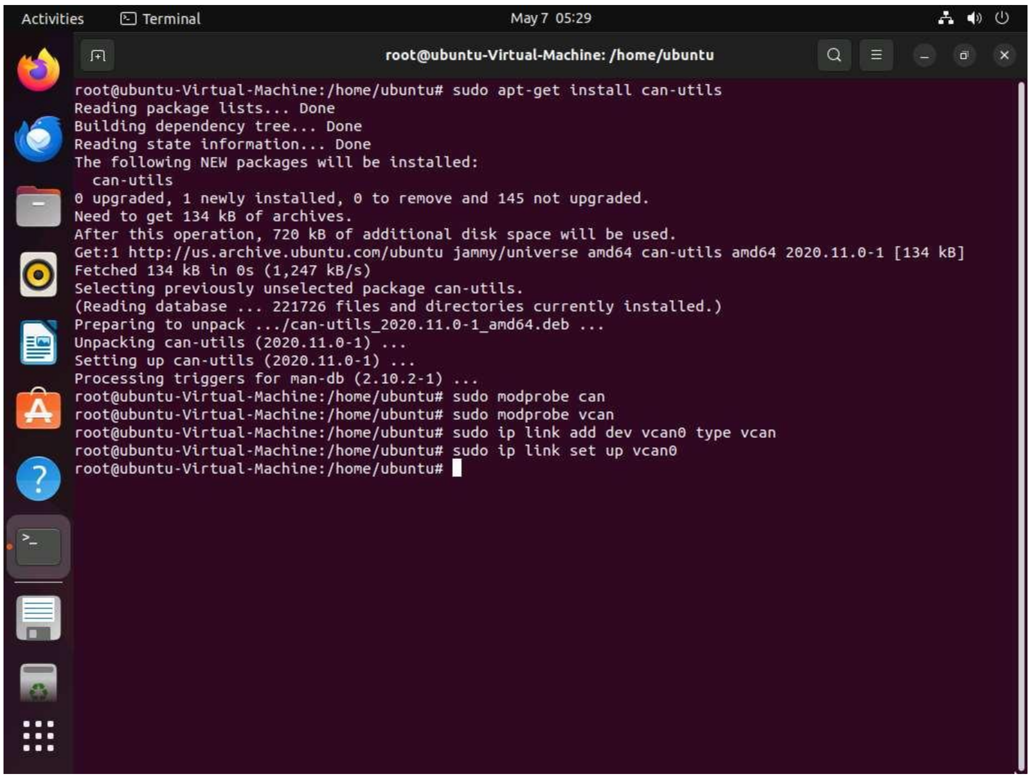
   While installing if prompted **Do you want to continue?**, type **Y** and press **Enter**.



```
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo apt-get install can-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  can-utils
0 upgraded, 1 newly installed, 0 to remove and 145 not upgraded.
Need to get 134 kB of archives.
After this operation, 720 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 can-utils amd64 2020.11.0-1 [134 kB]
Fetched 134 kB in 0s (1,247 kB/s)
Selecting previously unselected package can-utils.
(Reading database ... 221726 files and directories currently installed.)
Preparing to unpack .../can-utils_2020.11.0-1_amd64.deb ...
Unpacking can-utils (2020.11.0-1) ...
Setting up can-utils (2020.11.0-1) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu-Virtual-Machine:/home/ubuntu#
```

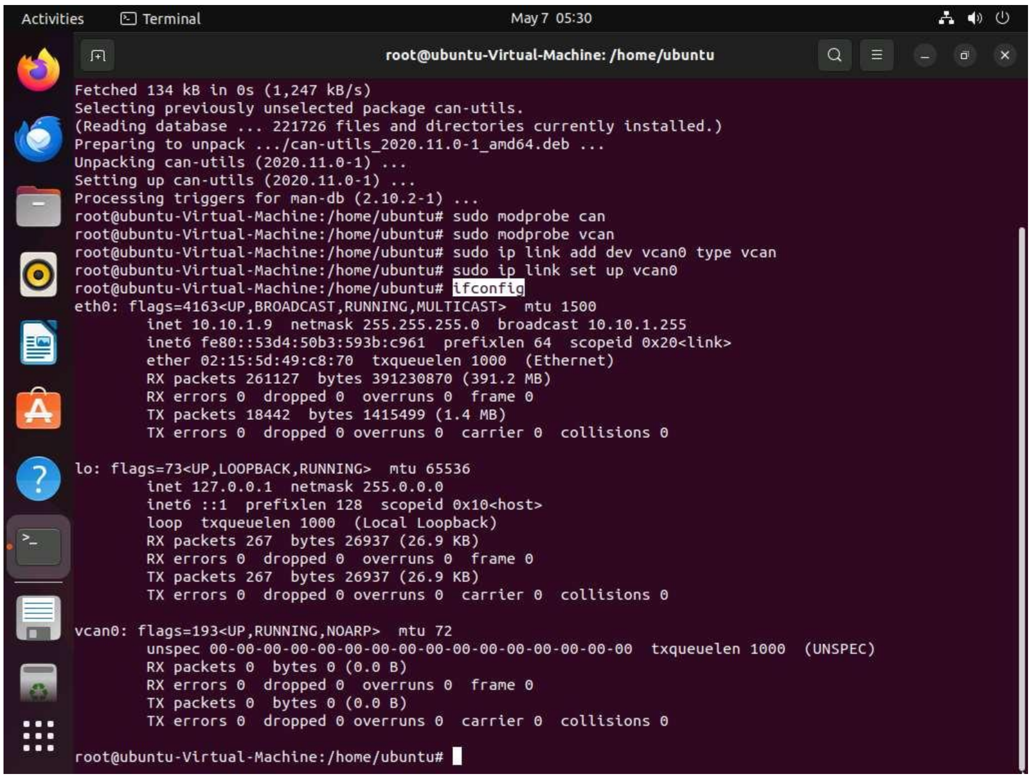
4. Now, to setup a virtual CAN interface issue following commands:

    - **sudo modprobe can**
    - **sudo modprobe vcan**
    - **sudo ip link add dev vcan0 type vcan**
    - **sudo ip link set up vcan0**



```
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo apt-get install can-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  can-utils
0 upgraded, 1 newly installed, 0 to remove and 145 not upgraded.
Need to get 134 kB of archives.
After this operation, 720 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 can-utils amd64 2020.11.0-1 [134 kB]
Fetched 134 kB in 0s (1,247 kB/s)
Selecting previously unselected package can-utils.
(Reading database ... 221726 files and directories currently installed.)
Preparing to unpack .../can-utils_2020.11.0-1_amd64.deb ...
Unpacking can-utils (2020.11.0-1) ...
Setting up can-utils (2020.11.0-1) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo modprobe can
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo modprobe vcan
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo ip link add dev vcan0 type vcan
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo ip link set up vcan0
root@ubuntu-Virtual-Machine:/home/ubuntu#
```

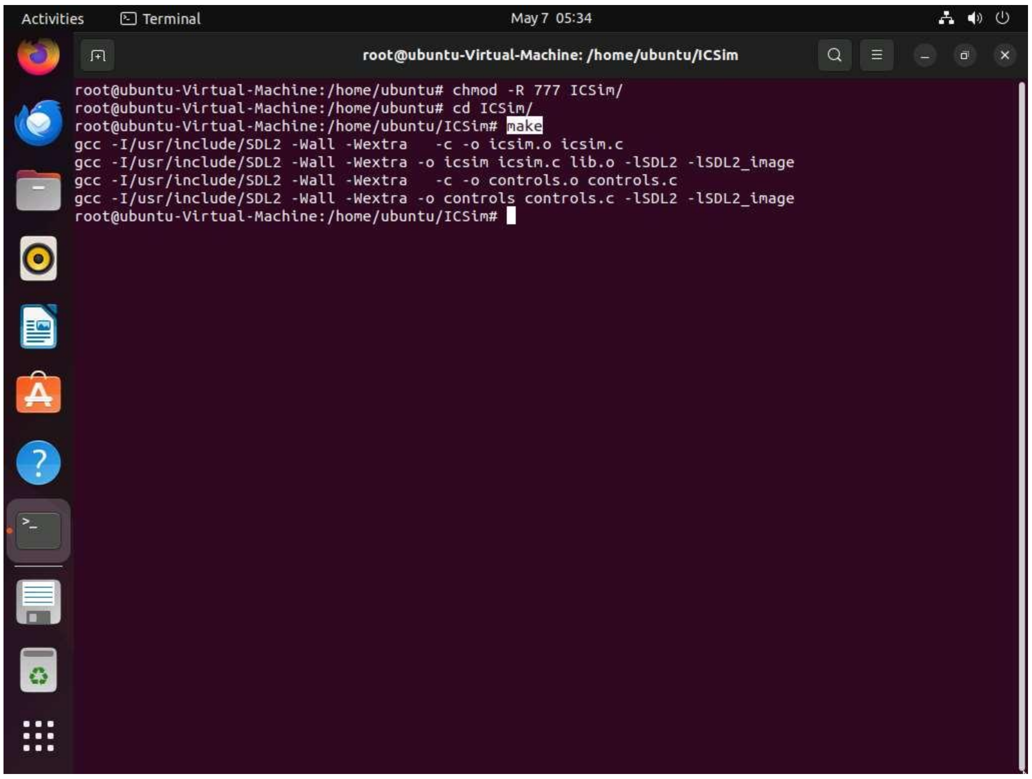
5. To check whether Virtual CAN interface is setup successfully, run **ifconfig**. Here, **vcan0** interface is present which confirms that our Virtual CAN interface is setup successfully.



```
Fetched 134 kB in 0s (1,247 kB/s)
Selecting previously unselected package can-utils.
(Reading database ... 221726 files and directories currently installed.)
Preparing to unpack .../can-utils_2020.11.0-1_amd64.deb ...
Unpacking can-utils (2020.11.0-1) ...
Setting up can-utils (2020.11.0-1) ...
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo modprobe can
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo modprobe vcan
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo ip link add dev vcan0 type vcan
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo ip link set up vcan0
root@ubuntu-Virtual-Machine:/home/ubuntu# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.1.9  netmask 255.255.255.0  broadcast 10.10.1.255
        inet6 fe80::53d4:50b3:593b:c961  prefixlen 64  scopeid 0x20<link>
        ether 02:15:5d:49:c8:70  txqueuelen 1000  (Ethernet)
        RX packets 261127  bytes 391230870 (391.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18442  bytes 1415499 (1.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 267  bytes 26937 (26.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 267  bytes 26937 (26.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vcan0: flags=193<UP,RUNNING,NOARP>  mtu 72
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu-Virtual-Machine:/home/ubuntu# 
```

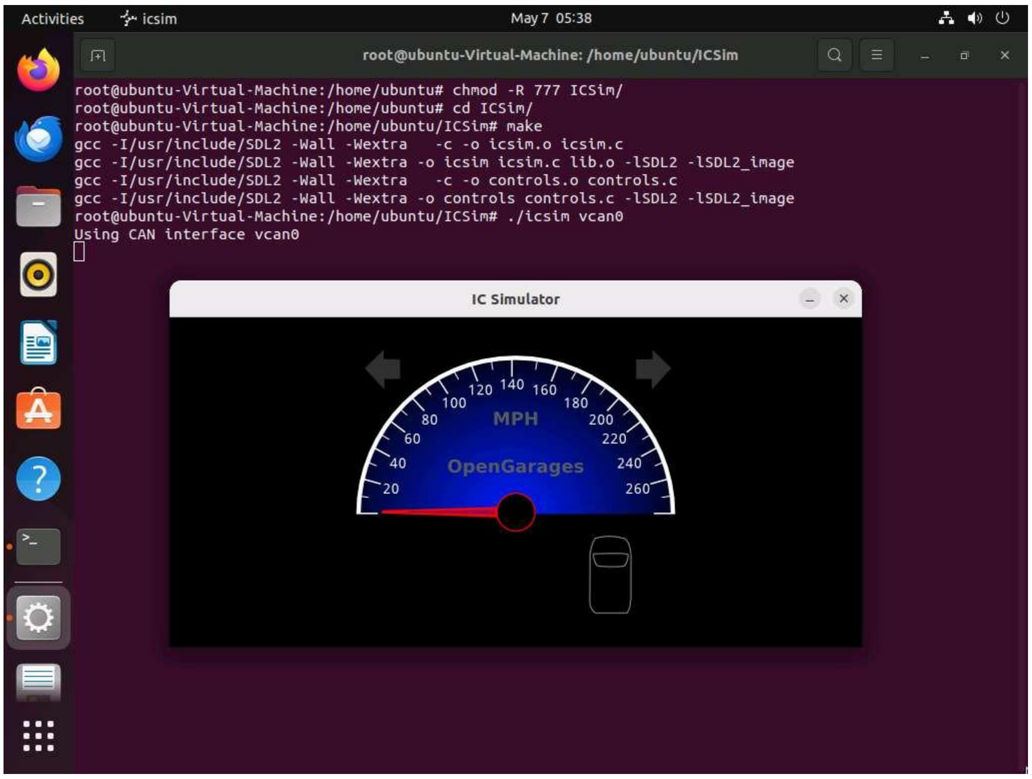
6. Run **chmod -R 777 ICSim** to give permissions to the ICSim folder.

7. Now, run **cd ICSim** to navigate to ICSim directory and execute **make** command to create two executable files for IC Simulator and CANBus Control Panel.



```
root@ubuntu-Virtual-Machine:/home/ubuntu# chmod -R 777 ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# make
gcc -I/usr/include/SDL2 -Wall -Wextra   -c -o icsim.o icsim.c
gcc -I/usr/include/SDL2 -Wall -Wextra -o icsim icsim.c lib.o -lSDL2 -lSDL2_image
gcc -I/usr/include/SDL2 -Wall -Wextra   -c -o controls.o controls.c
gcc -I/usr/include/SDL2 -Wall -Wextra -o controls controls.c -lSDL2 -lSDL2_image
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# 
```

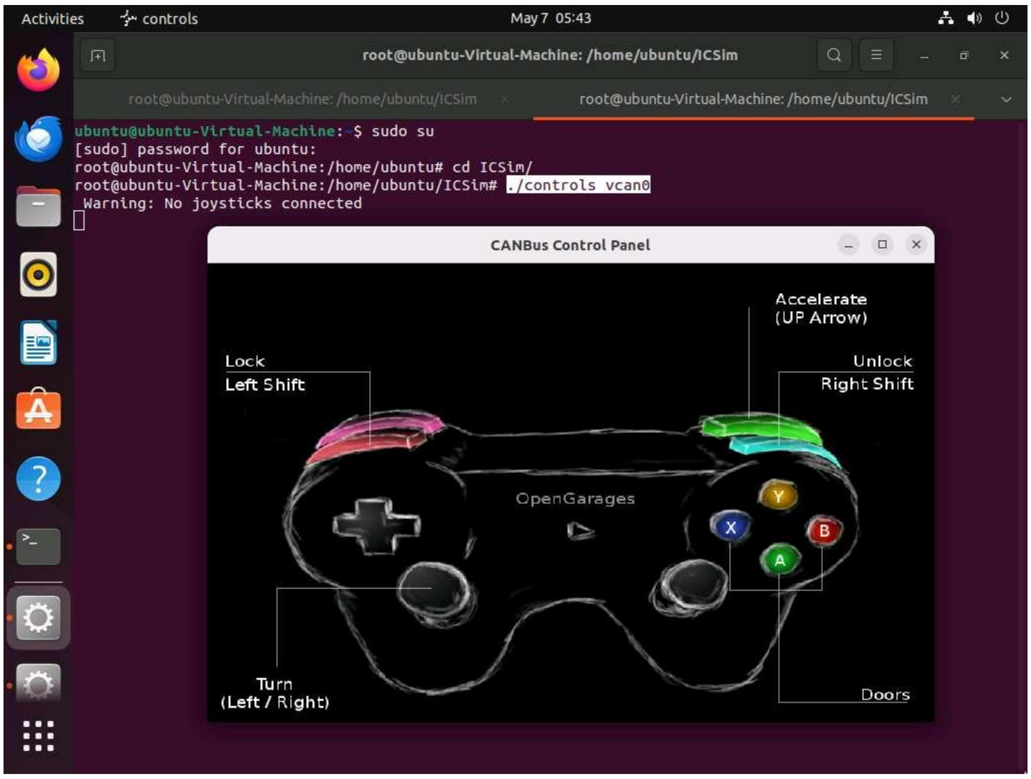
8. Run **./icsim vcan0** to start the ICSim simulator. You will see the IC Simulator interface as shown in the screenshot.



```
root@ubuntu-Virtual-Machine:/home/ubuntu# chmod -R 777 ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# make
gcc -I/usr/include/SDL2 -Wall -Wextra   -c -o icsim.o icsim.c
gcc -I/usr/include/SDL2 -Wall -Wextra -o icsim icsim.c lib.o -lSDL2 -lSDL2_image
gcc -I/usr/include/SDL2 -Wall -Wextra   -c -o controls.o controls.c
gcc -I/usr/include/SDL2 -Wall -Wextra -o controls controls.c -lSDL2 -lSDL2_image
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# ./icsim vcan0
Using CAN interface vcan0
```

9. Open a new terminal tab and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**). Navigate to ICSim directory to do so run **cd ICSim/**.

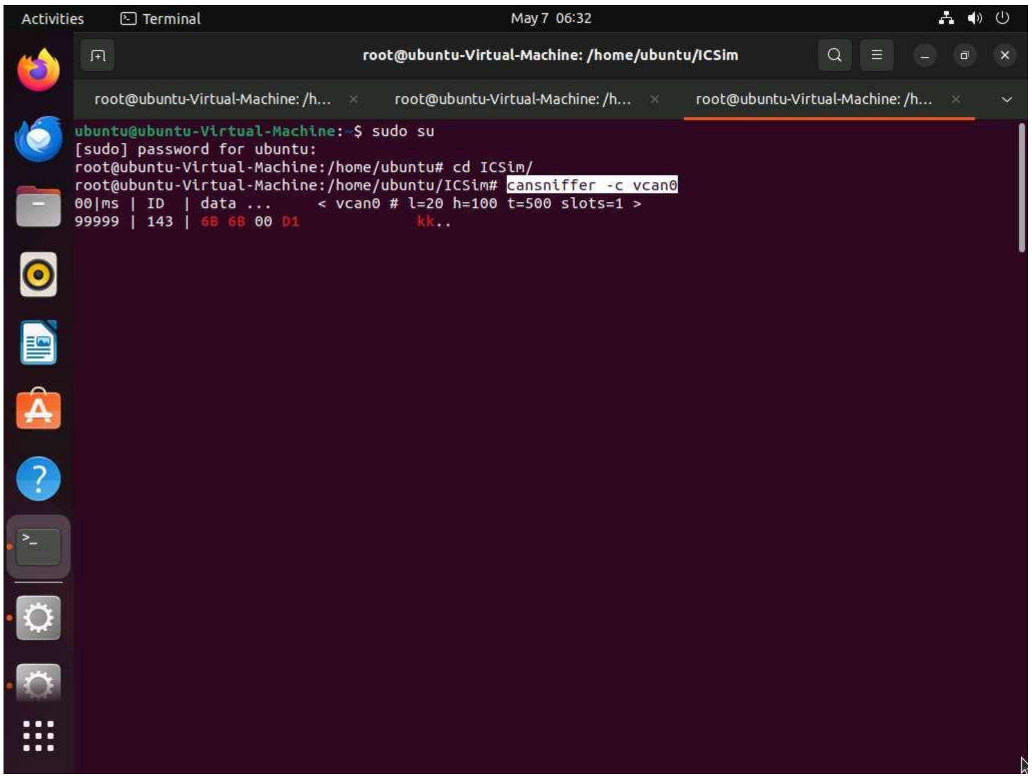
10. Execute **./controls vcan0** to start the CANBus Control Panel. You will see the CANBus Control Panel interface as shown in the screenshot.



```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# ./controls vcan0
 Warning: No joysticks connected
```

11. Now, we will start sniffer to capture the traffic sent to the ICSim Simulator by CANBus control panel simulator. To do so, open a new terminal tab and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**). Navigate to ICSim directory to do so run **cd ICSim/**.

12. Execute **cansniffer -c vcan0** to start sniffing on the vcan0 interface. Leave this sniffer on.


Activities
Terminal
May 7 06:32
root@ubuntu-Virtual-Machine: /home/ubuntu/ICSim
root@ubuntu-Virtual-Machine: /h...
root@ubuntu-Virtual-Machine: /h...
root@ubuntu-Virtual-Machine: /h...

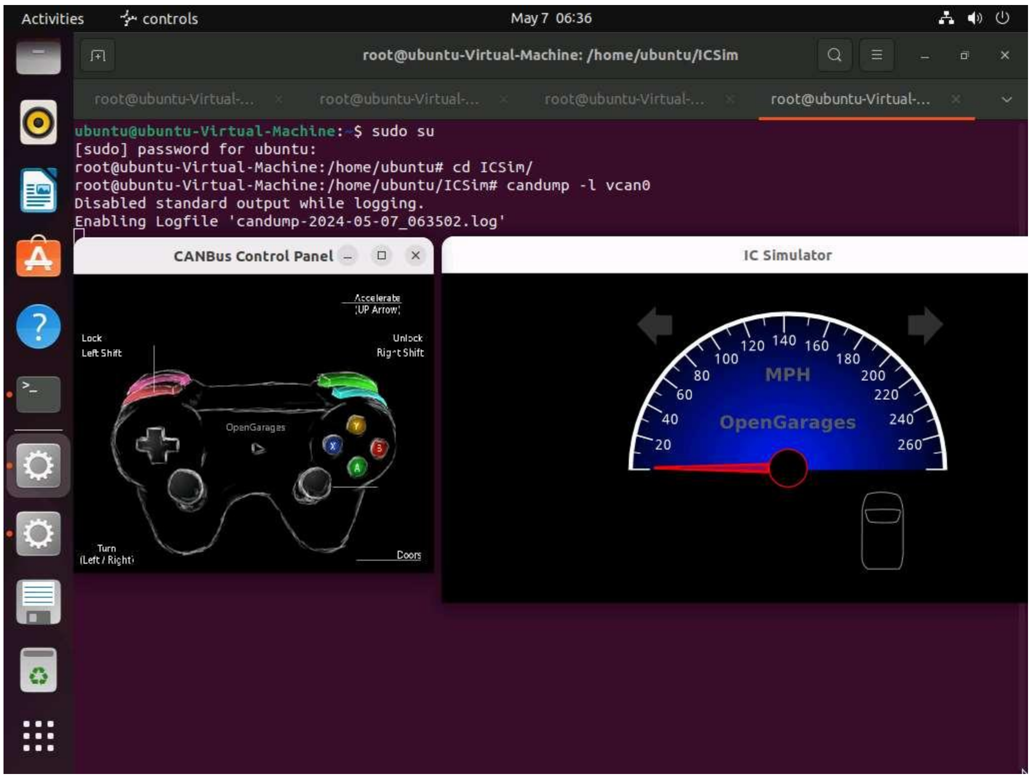
```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# cansniffer -c vcan0
00|ms | ID  | data ...     < vcan0 # l=20 h=100 t=500 slots=1 >
99999 | 143 | 6B 6B 00 D1             kk..
```

13. Open a new terminal and execute **sudo su** to run the programs as a root user (When prompted, enter the password toor). Navigate to ICSim directory to do so run **cd ICSim/**. To capture the logs run **candump -l vcan0**.

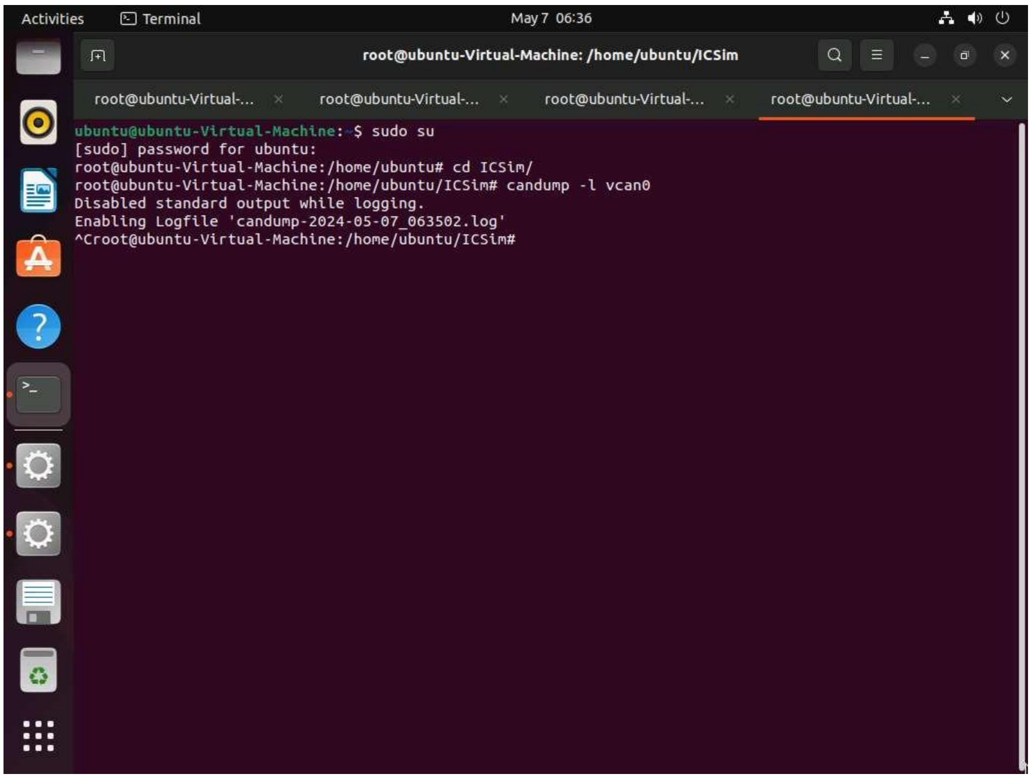
14. After starting to capture the logs, open ICSim and Controller simulator and perform functions such as acceleration, turning left/right, opening and locking doors so that logs are generated. Once you are done, terminate the ongoing process by pressing **Ctrl + C**.

    Use the following keys to perform various functions

ICSim Functions	**Keys**
Accelerate	Up arrow
Left/Right Turn	Left arrow/ Right arrow
Unlock Rear Left/Right doors	Right Shift + X / Right Shift + Y
Unlock Front Left/Right doors	Right Shift +A / Right Shift + B
Lock all doors	Hold Right Shift key + Tap Left Shift
Unlock all doors	Hold Left Shift key + Tap Right Shift

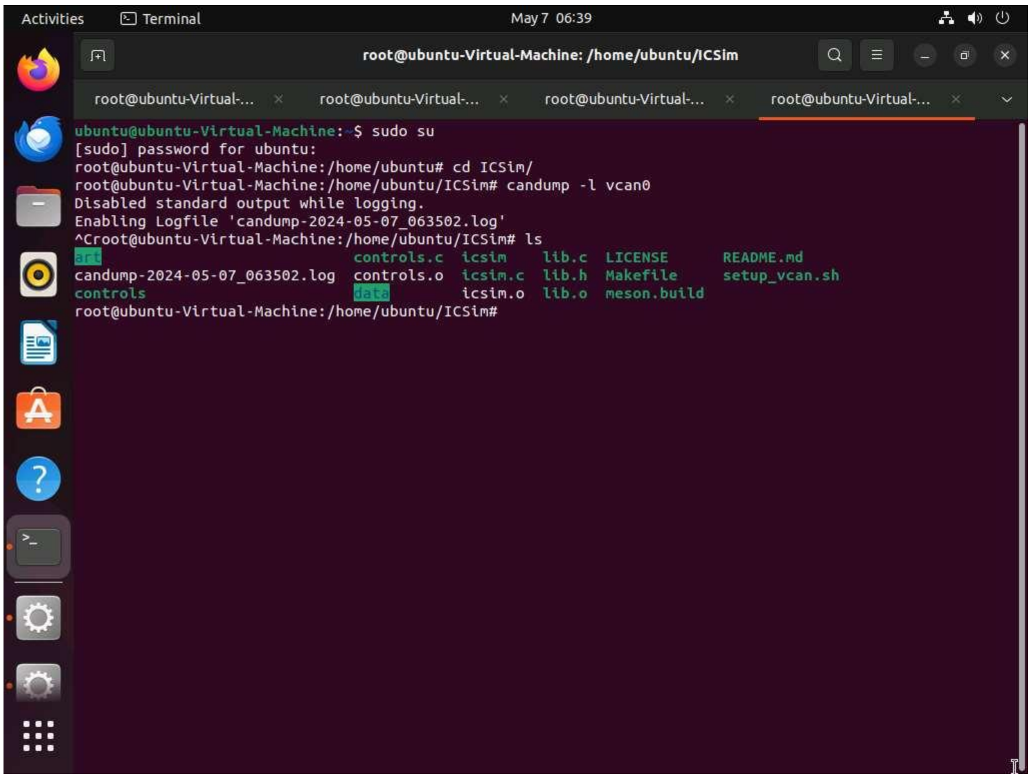


```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# candump -l vcan0
Disabled standard output while logging.
Enabling Logfile 'candump-2024-05-07_063502.log'
```



```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# candump -l vcan0
Disabled standard output while logging.
Enabling Logfile 'candump-2024-05-07_063502.log'
^Croot@ubuntu-Virtual-Machine:/home/ubuntu/ICSim#
```

15. Now verify if you have obtained the log file by executing **ls** command.
    The **.log** file has been generated as shown in the screenshot.

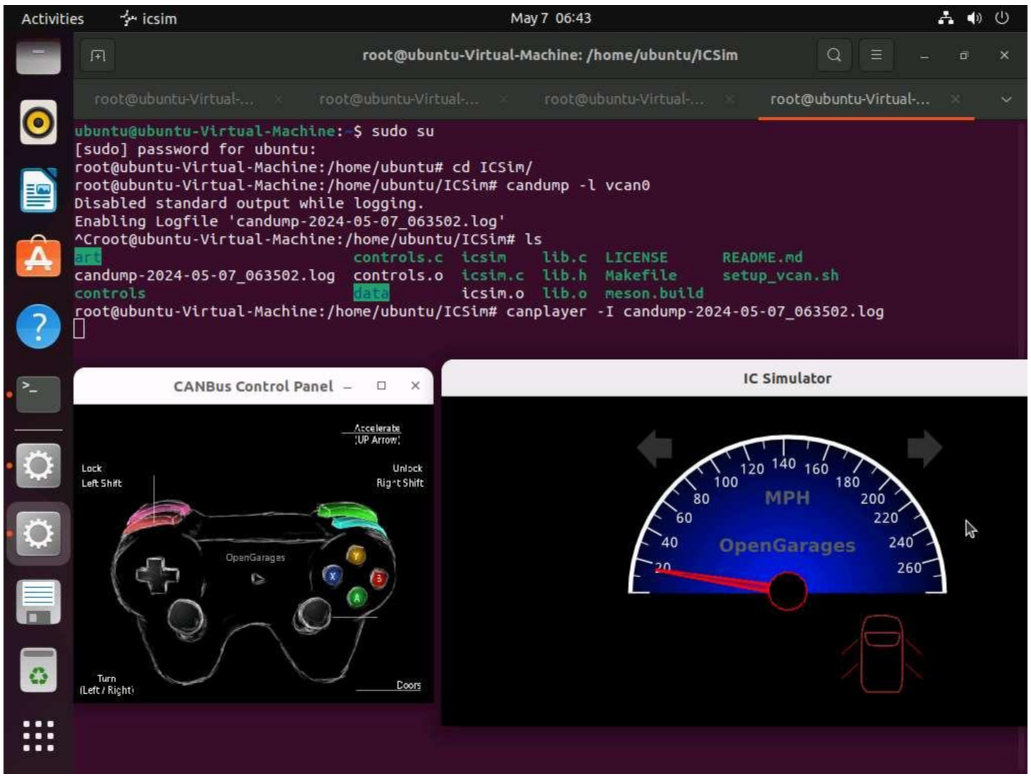


```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# candump -l vcan0
Disabled standard output while logging.
Enabling Logfile 'candump-2024-05-07_063502.log'
^Croot@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# ls
art                             controls.c  icsim    lib.c  LICENSE      README.md
candump-2024-05-07_063502.log   controls.o  icsim.c  lib.h  Makefile     setup_vcan.sh
controls                        data        icsim.o  lib.o  meson.build
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim#
```

16. Now, to perform replay attack, run **canplayer -I candump-2024-05-07_063502.log** and press enter.

    Once the log file is executed, you can see the movements that were performed while creating the log file in real time in IC Simulator and CANBus control panel simulator.

    The log file name might vary while performing lab.



```
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# candump -l vcan0
Disabled standard output while logging.
Enabling Logfile 'candump-2024-05-07_063502.log'
^Croot@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# ls
art                            controls.c  icsim    lib.c  LICENSE      README.md
candump-2024-05-07_063502.log  controls.o  icsim.c  lib.h  Makefile     setup_vcan.sh
controls                       data        icsim.o  lib.o  meson.build
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# canplayer -I candump-2024-05-07_063502.log
```

17. This concludes the demonstration of performing replay attack to exploit CAN protocol.

18. Close all open windows and document all the acquired information.

Question 18.3.1.1

In Ubuntu machine install ICSim simulator, start a CAN sniffer and perform functions such as acceleration, turning left/right, opening and locking doors in the simulator to generate the logs. Perform replay attack using the sniffed log file. Enter the interface that is used while sniffing the can traffic in Ubuntu.