

Loading your lab content

Close Window

1

---

Close

2

3

- Reconnect
- Power On
- Pause
- Resume
- Reset/Reboot
- Power Off
- Fit Window to Machine
- Fit Machine To Window
- Open in New Window
- Split Windows
- Revert Machine
- Reset Internet Gateway

4

- Ctrl+Alt+Delete
- ALT+Tab
- Windows Key
  - Windows Key
  - Windows Key + D
  - Windows Key + E
  - Windows Key + F
  - Windows Key + M
  - Windows Key + R
  - Windows Key + X
  - Windows Key + ...
- Windows Key
- Type Text
  - Type Username
  - Type Password
  - Type Clipboard Text
- Virtual Keyboard

Windows 11<sup>5</sup>

Windows 11  
Windows Server 2022  
Windows Server 2019  
Parrot Security  
Ubuntu

## Poor Connection

---

Full Screen  
Power and Display  
Keyboard  
Machine Selection

This machine must be controlled outside of your browser via Remote Desktop.

Launch Remote Desktop

Username:

Password:

The selected machine is off.

Start

Machine is open in a separate window. [Close Window](#)

X

- Esc
- F1
  - F2
  - F3
  - F4
  - F5
  - F6
  - F7
  - F8
  - F9
  - F10
  - F11
  - F12
  - PrtSc
  - ScrLk
  - Pause
  - `
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
  - 8
  - 9
  - 0
  - -
  - =
  - ← Backspace
  - Insert

- Home
- P Up

- NLock

- /
- \*
- -
- Tab
- q
- w
- e
- r
- t
- y
- u
- i
- o
- p
- [
- ]
- \
- Delete
- End
- P Down

- 7

- 8
- 9
- +
- Caps
- a
- s
- d
- f
- g
- h
- j
- k
- l
- ;
- '
- ↵ Enter

- 4

- 5
- 6
- Shift
- z
- x
- c

- v
- b
- n
- m
- ,
- .
- /
- Shift
- ↑

- 1

- 2
- 3
- Enter
- Ctrl
- Win
- Alt
- Alt
- Win
- Ctrl
- ←
- ↓
- →

- 0

- .

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

6

Password

7

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

- Not Connected

Username

Password

Reconnect

IoT and OT Hacking<sup>8</sup>

[Exit Lab](#)

Save Progress And Exit

End Lab

[InstructionsResources](#)

## Module 18: IoT and OT Hacking

### Scenario

---

Type Text

Type Text

IoT and OT Hacking

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
  - o Gather information using online footprinting tools
2. Capture and analyze IoT device traffic
  - o Capture and analyze IoT traffic using Wireshark
3. Perform IoT Attacks
  - o 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.

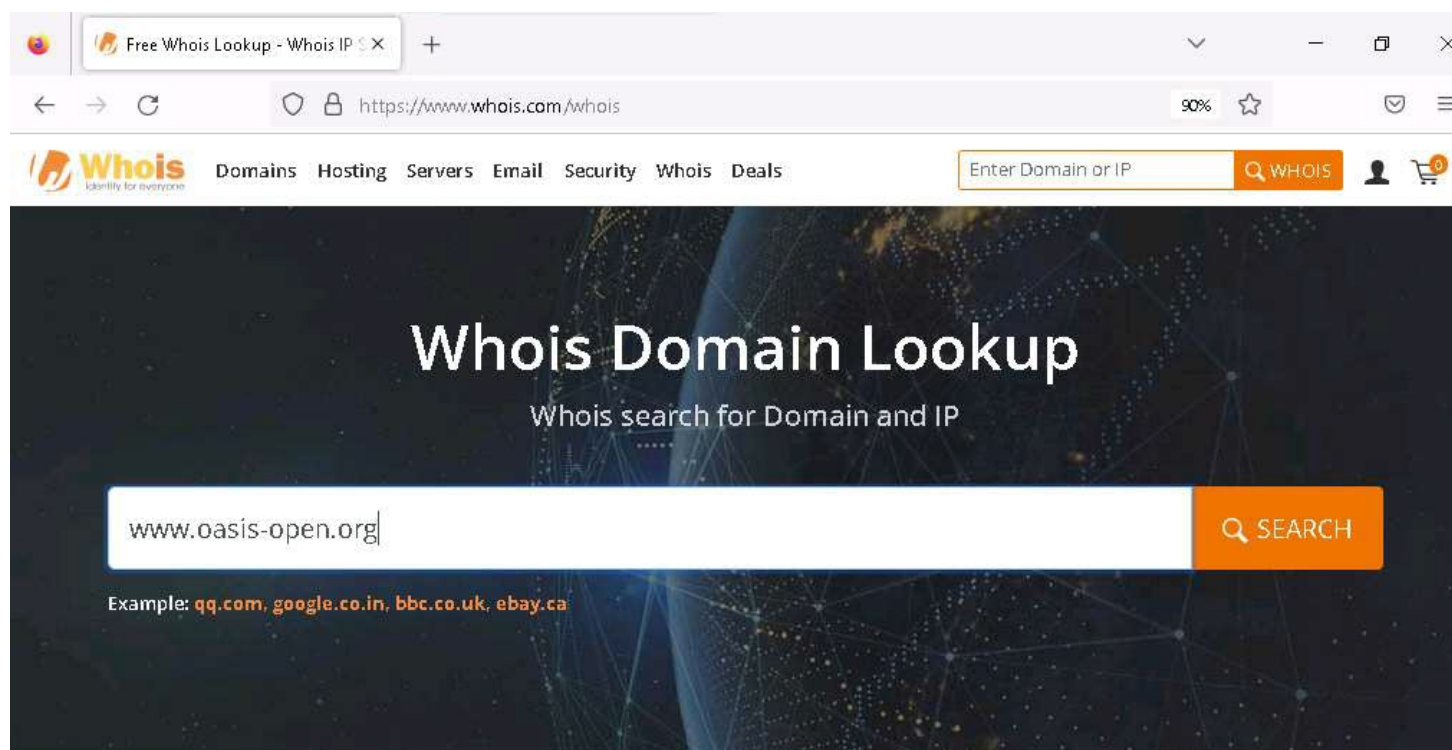
### Task 1: Gather Information using Online Footprinting Tools

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**.
2. Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.
3. Launch any web browser, go to <https://www.whois.com/whois> (here, we are using **Mozilla Firefox**).
4. The **Whois Domain Lookup** page appears; type **www.oasis-open.org** in the search field and click **SEARCH**.
5. 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.
- 6.



## Frequently Asked Questions

+ What is a Whois domain lookup?

File Explorer



7. The result appears, displaying the following information, as shown in the screenshots: Domain Information, Registrant Contact, and Raw Whois Data.
8. 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.
- 9.

Whois oasis-open.org

https://www.whois.com/whois/oasis-open.org

Whois identity for everyone

Domains Hosting Servers Email Security Whois Deals

Enter Domain or IP Q WHOIS

### 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...

On Sale!

.LIFE

.LIFE @ \$2.48 \$35.88

Introducing

WORDPRESS HOSTING

\$5.48 /mo

VIEW MORE

12:31 AM 3/14/2024

10. Whois lookup reveals available information on a hostname, IP address, or domain.
11. Now, open a new tab, and go to <https://www.exploit-db.com/google-hacking-database>.
12. The **Google Hacking Database** page appears; type **SCADA** in the **Quick Search** field and press **Enter**.
13. The result appears, which displays the Google dork related to SCADA, as shown in the screenshot.



14.

The screenshot shows the Exploit Database website with the Google Hacking Database search results for the query 'SCADA'. The results are displayed in a table with columns for Date Added, Dork, Category, and Author. The table contains 7 entries, which are highlighted with a red border. The search bar at the top right shows the query 'SCADA'. The page indicates that 7 entries are shown, filtered from 7,915 total entries.

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

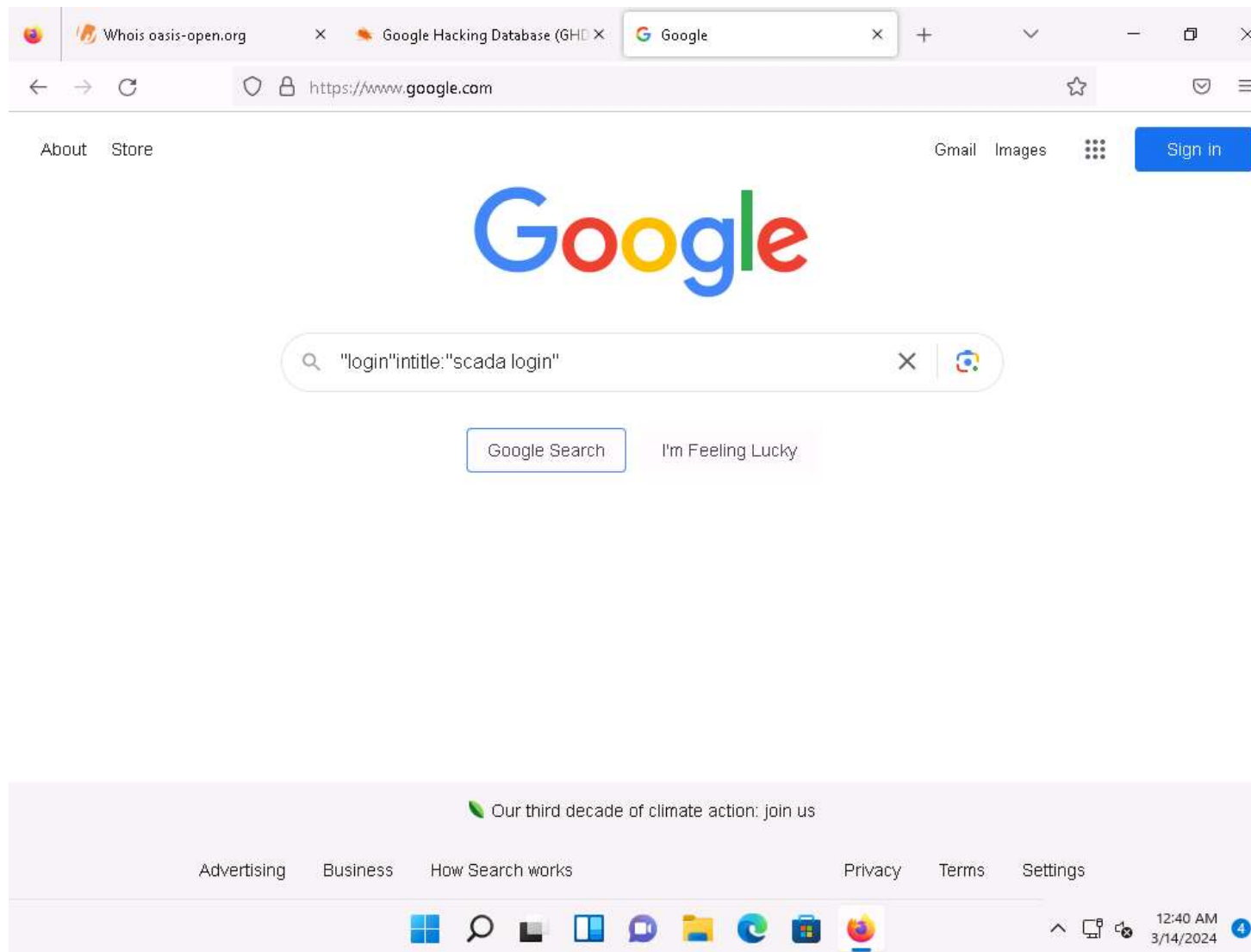
Showing 1 to 7 of 7 entries (filtered from 7,915 total entries)

FIRST PREVIOUS 1 NEXT LAST

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

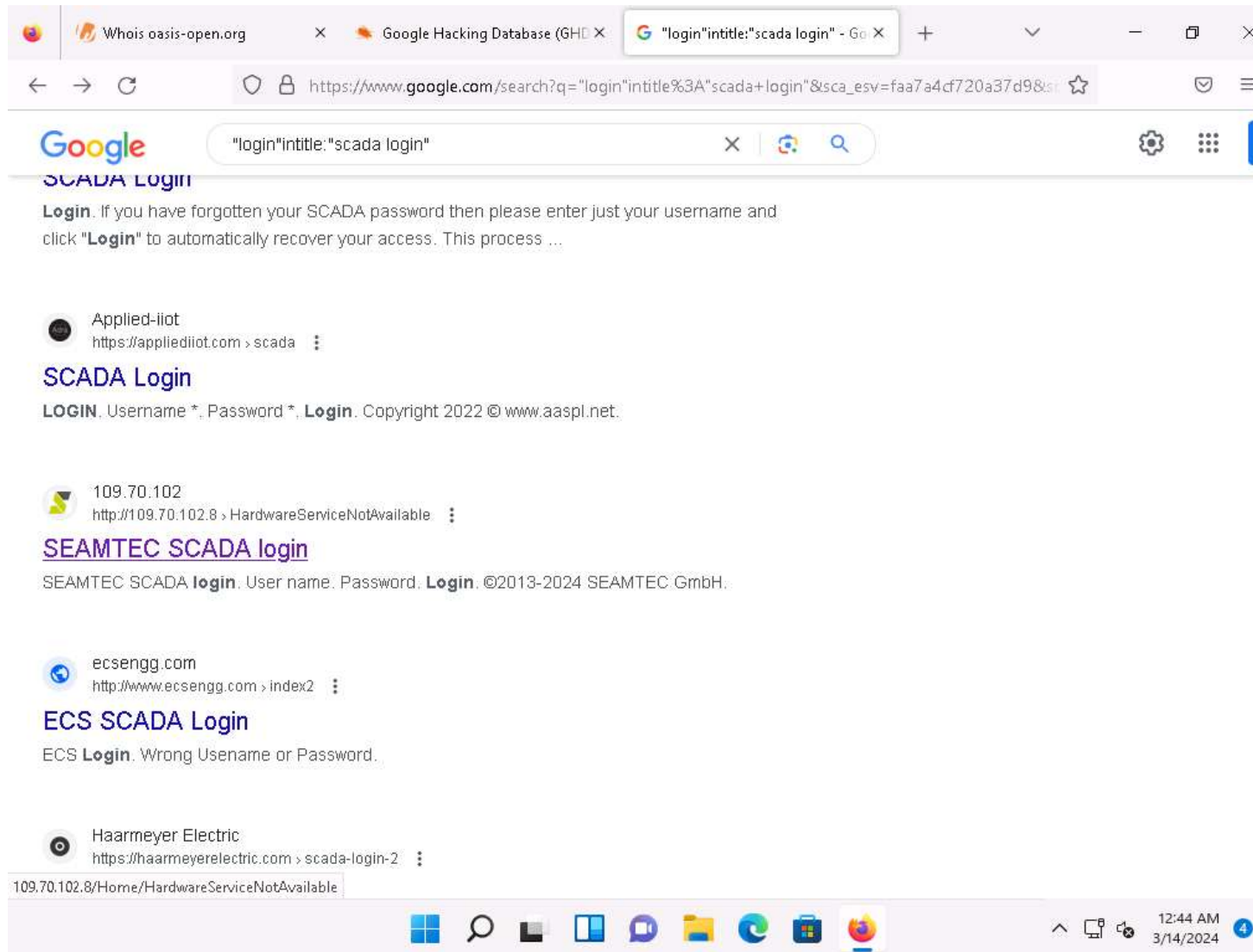
16. Open a new tab and go to <https://www.google.com>. In the search field, enter **"login" intitle:"scada login"**.

17.



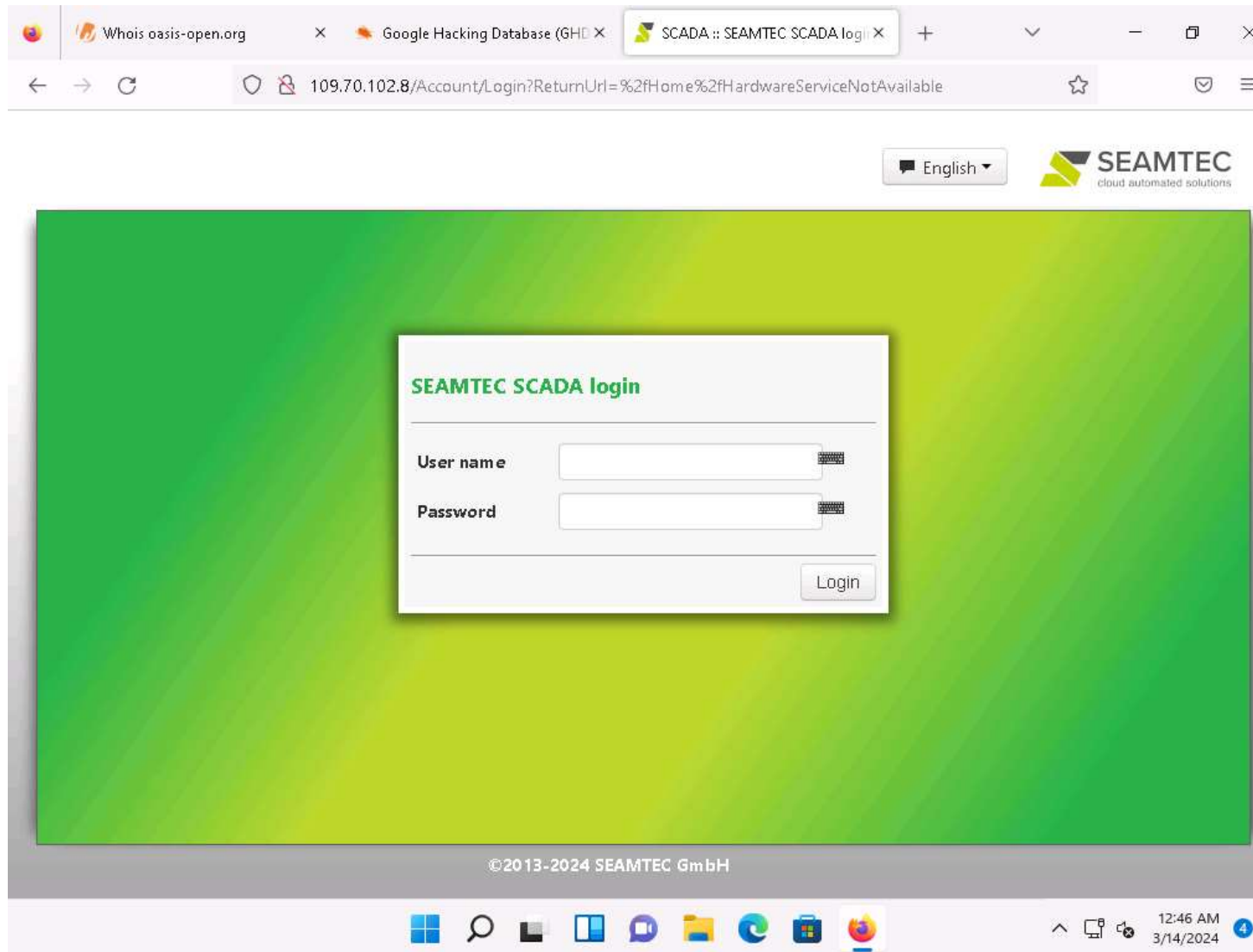
18. The search result appears; click any link (here, **SEAMTEC SCADA login**).

19.



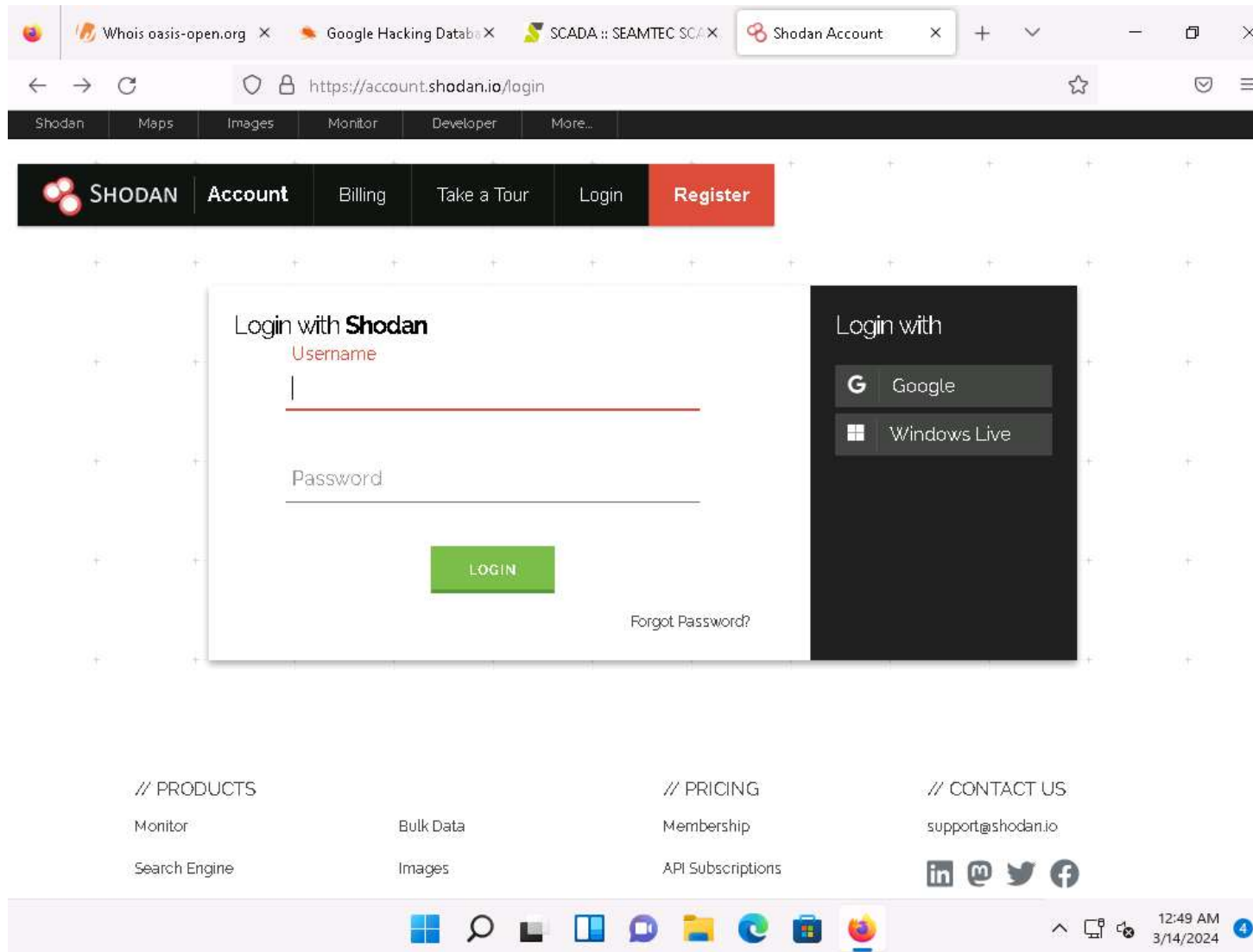
20. Advanced Google hacking refers to the art of creating complex search engine queries by employing advanced Google operators to extract sensitive or hidden information about a target company from the Google search results.
21. The **SEAMTEC SCADA login** page appears, as shown in the screenshot.
22. In the login form, you can brute-force the credentials to gain access to the target SCADA system.

23.



24. Similarly, you can use advanced search operators such as **intitle:"index of" scada** to search sensitive SCADA directories that are exposed on sites.
25. Now, in the browser window, open a new tab and go to **<https://account.shodan.io/login>**.
26. The **Login with Shodan** page appears; enter your username and password in the **Username** and **Password** fields, respectively; and click **Login**.
27. If you do not have an existing account, then go to the **Register** option to register yourself .

28.



29. 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**.
30. If the **Would you like Firefox to save this login for shodan.io?** notification appears, click **Don't Save**.
31. The **Shodan** main page appears; type **port:1883** in the address bar and press **Enter**.
32. Port 1883 is the default MQTT port; 1883 is defined by IANA as MQTT over TCP.

33.

Whois oasis-open.org X Google Hacking Database X SCADA :: SEAMTEC SCADA X Shodan Search Engine X

https://account.shodan.io/login

Shodan Maps Images Monitor Developer More...

SHODAN Explore Downloads Pricing port:1883 Account

## 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

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

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

36.

The screenshot shows the Shodan Search interface. The browser tabs include 'Whois oasis-open.org', 'Google Hacking Database', 'SCADA :: SEAMTEC SCADA', and 'port:1883 - Shodan Search'. The address bar shows 'https://account.shodan.io/login'. The main content area displays search results for IP addresses. The sidebar on the left shows filters for '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), 'TOP ORGANIZATIONS' (SK Broadband Co: 357,215, Google LLC: 355,579, Aliyun Computing C: 40,166, Flyio, Inc.: 22,508, Huawei Public Clou: 10,559), and 'TOP PRODUCTS'. The main results pane shows details for four IP addresses: 34.49.29.35, 130.211.8.229, 213.188.219.148, and 39.125.233.41. Each entry includes the IP address, a timestamp, a status message ('No data returned'), and a list of associated hostnames and organizations. The bottom of the screen shows the Windows taskbar with the time 9:44 PM on 3/14/2024.

IP Address	Timestamp	Status	Hostnames	Organizations
34.49.29.35	2024-03-15T04:43:20.440441	No data returned	35.29.49.34.bogoo gleusercontent.com	Google LLC United States, Kansas City
130.211.8.229	2024-03-15T04:43:13.001874	No data returned	229.8.211.130.bogoo gleusercontent.com	Google LLC United States, Kansas City
213.188.219.148	2024-03-15T04:43:00.029015	No data returned	Flyio, Inc.	United States, Chicago
39.125.233.41	2024-03-15T04:42:48.171868	MQTT Connection Code: @	SK Broadband Co Ltd	Korea, Republic of, Yeosu

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



38.

The screenshot shows the Shodan search engine interface. The top navigation bar includes links for Shodan, Maps, Images, Monitor, Developer, and More. The main header features the Shodan logo, navigation links (Explore, Downloads, Pricing), a search bar, and an Account link. The background is a satellite map of Kansas City. Below the map, there are buttons for 'Regular View' and 'Raw Data'. The search results are displayed in two columns. The left column, titled 'General Information', lists details for a specific host: Hostnames (blurred), Domains (GOOGLEUSERCONTENT.COM), Cloud Provider (Google), Cloud Region (global), Country (United States), City (Kansas City), and Organization (Google LLC). The right column, titled 'Open Ports', shows a grid of 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.

39. Similarly, you can gather additional information on a target device using the following Shodan filters:
  - o **Search for Modbus-enabled ICS/SCADA systems:**
  - o port:502
  - o **Search for SCADA systems using PLC name:**
  - o "Schneider Electric"
  - o **Search for SCADA systems using geolocation:**
  - o SCADA Country:"US"
40. 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.
41. 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.
42. 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?

Score

## 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

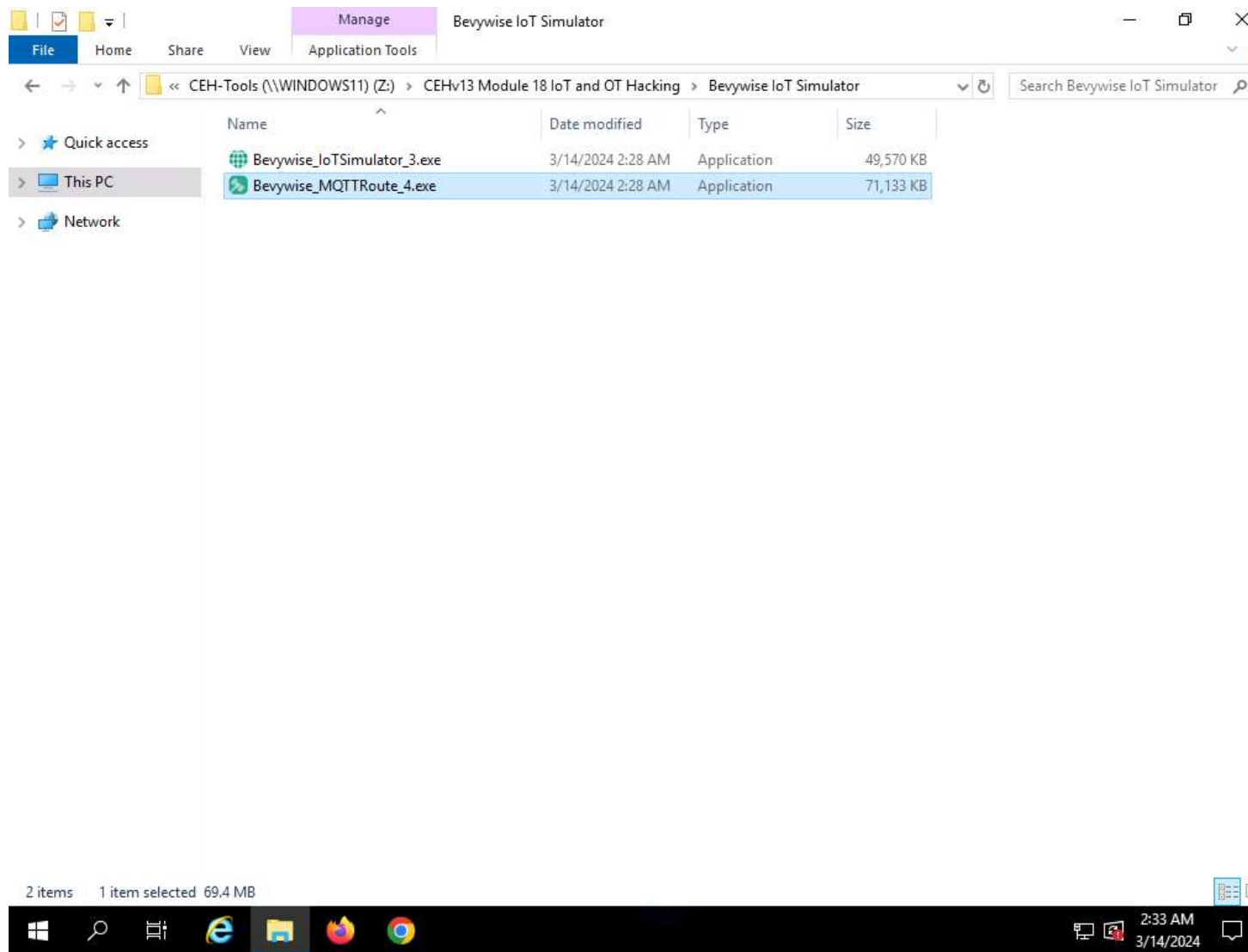
Wireshark is a free and open-source packet analyzer. It facilitates network troubleshooting, analysis, software and communications protocol development, and education. It is used to identify the target OS and sniff/capture the response generated from the target machine to the machine from which a request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

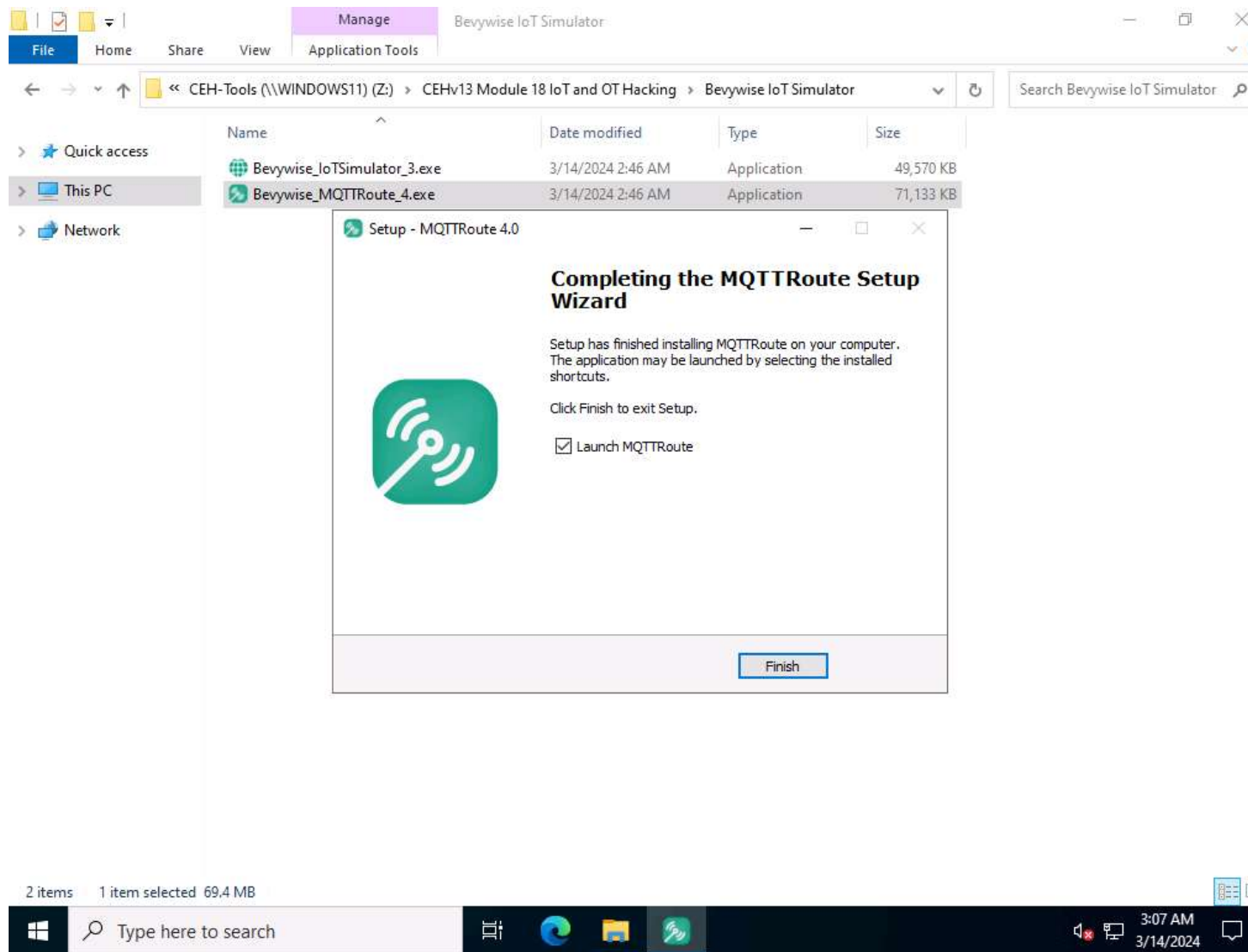
1. To install the **MQTT Broker** on the **Windows Server 2019**, click [Windows Server 2019](#) to launch **Windows Server 2019** machine. Click [Ctrl+Alt+Delete](#) and login with **Administrator/Pa\$\$w0rd**.
2. Navigate to **Z:\CEHv13 Module 18 IoT and OT Hacking\Bevywise IoT Simulator** folder and double-click on the **Bevywise\_MQTTRoute\_4.exe** file.

3.



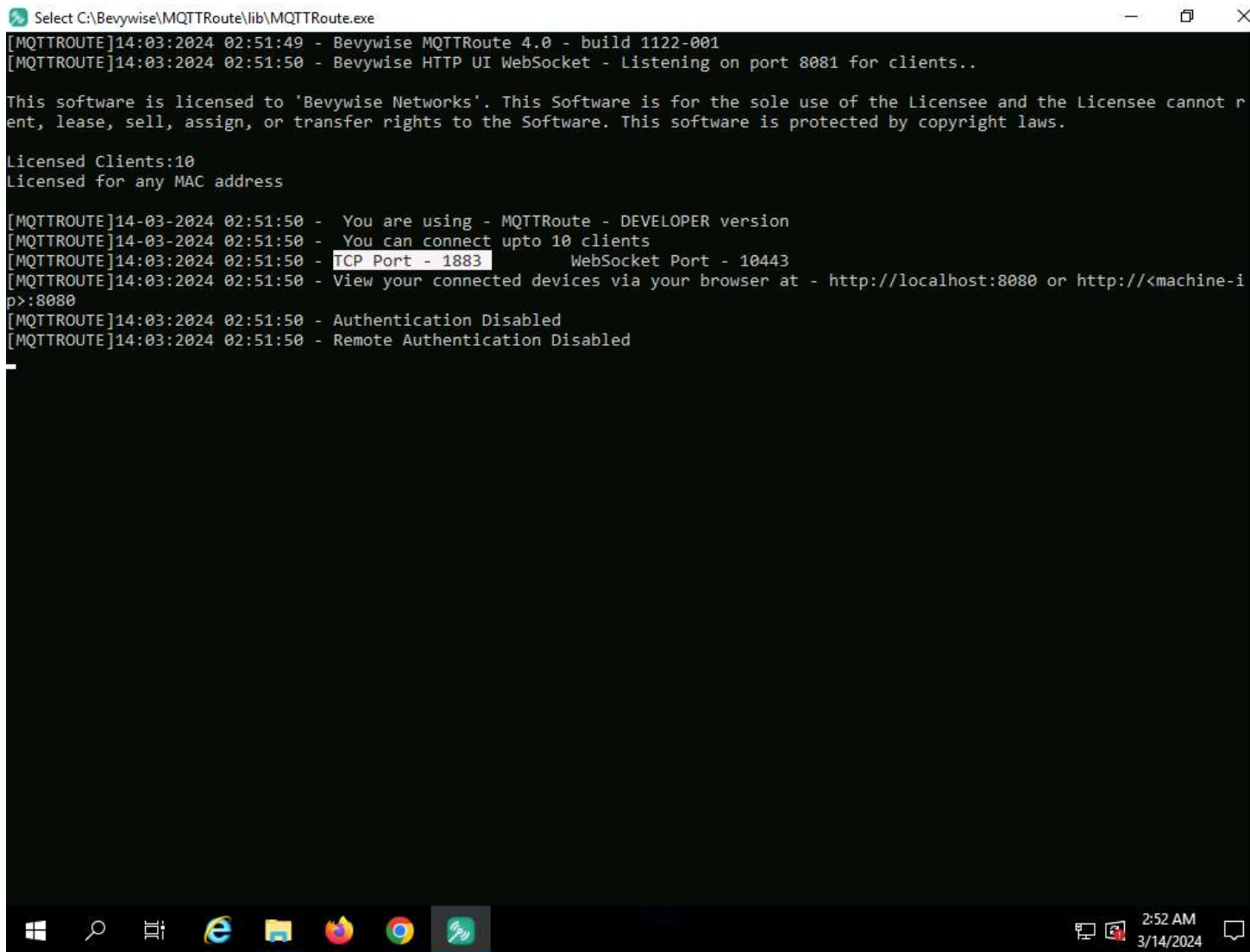
4. If **Open File - Security Warning** popup appears, click **Run**.
5. The **Setup - MQTTRoute 4.0** window opens. Select **I accept the agreement** and click on **Next**. Follow the wizard driven steps to install the tool.
6. After the installation completes, click on **Finish**. Ensure that **Launch MQTTRoute** is checked.

7.



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

9.



```
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..

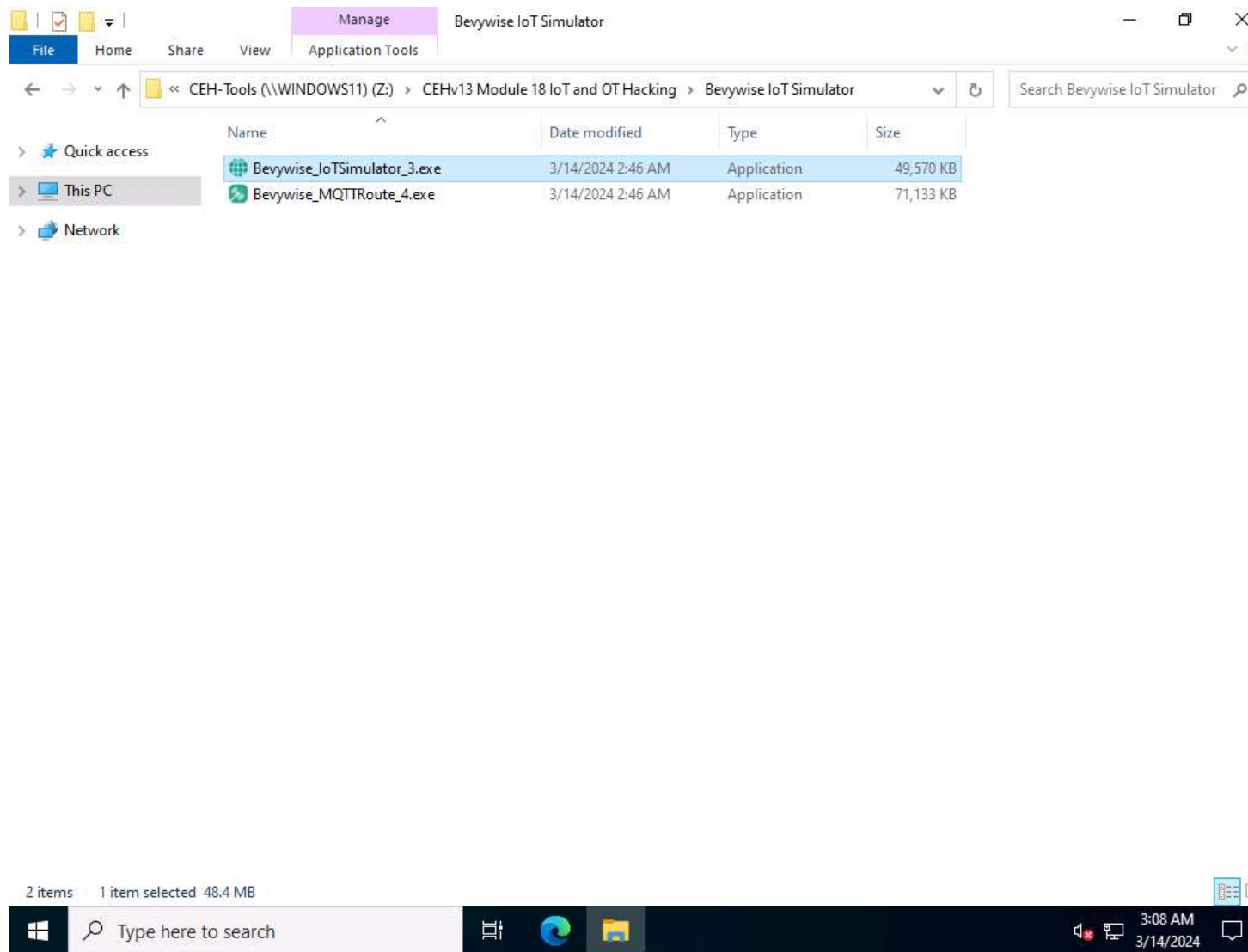
This software is licensed to 'Bevywise Networks'. This Software is for the sole use of the Licensee and the Licensee cannot r
ent, lease, sell, assign, or transfer rights to the Software. This software is protected by copyright laws.

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
```

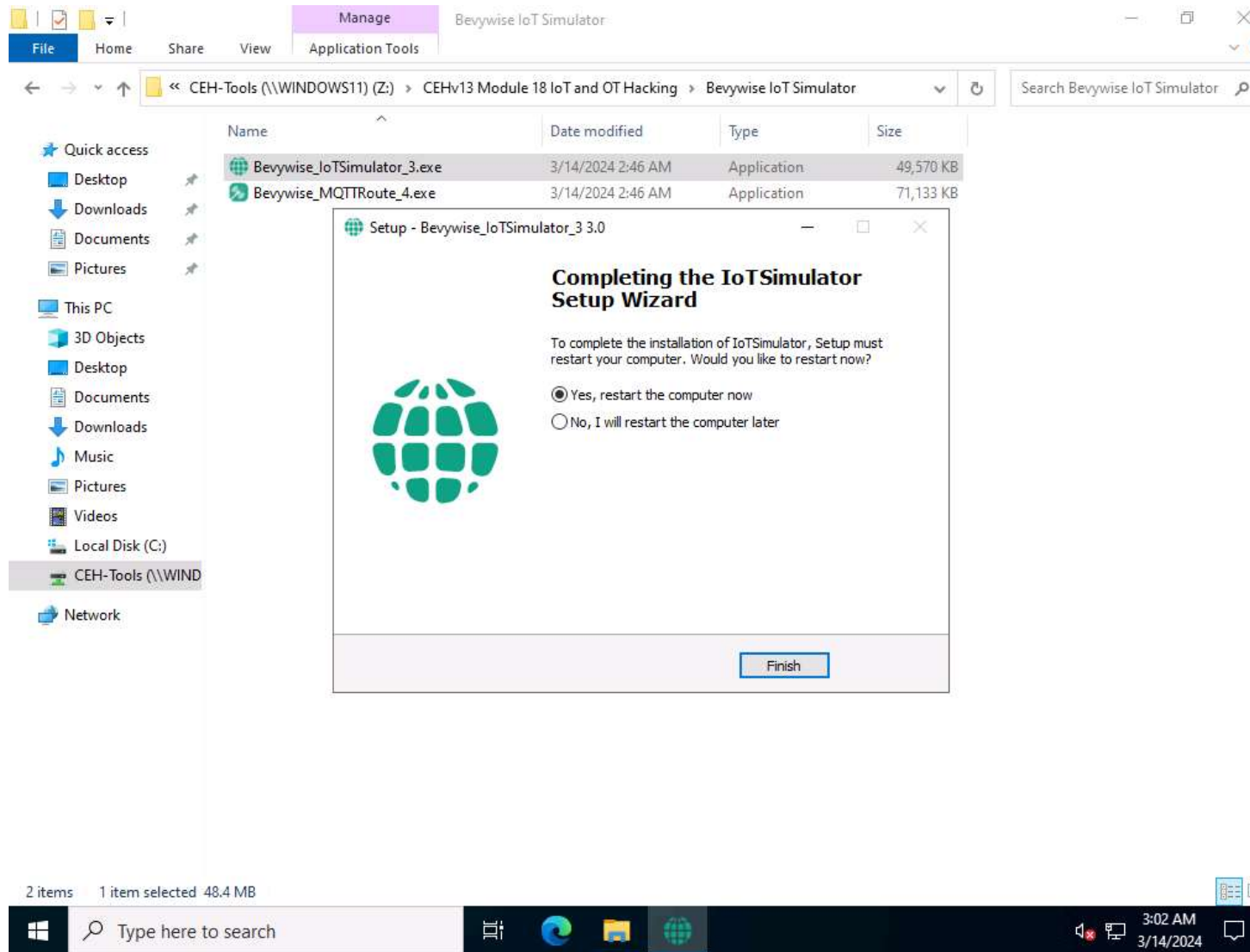
10. We have installed MQTT Broker successfully and leave the Bevywise MQTT **running**.
11. To create IoT devices, we must install the **IoT simulator** on the client machine.
12. Click [Windows Server 2022](#) to switch to **Windows Server 2022** machine. Click [Ctrl+Alt+Delete](#) and login with **Administrator/Pa\$\$w0rd**.
13. If the network screen appears, click **Yes**.
14. Navigate to **Z:\CEHv13 Module 18 IoT and OT Hacking\Bevywise IoT Simulator** folder and double-click on the **Bevywise\_IoTSimulator\_3.exe** file.

15.



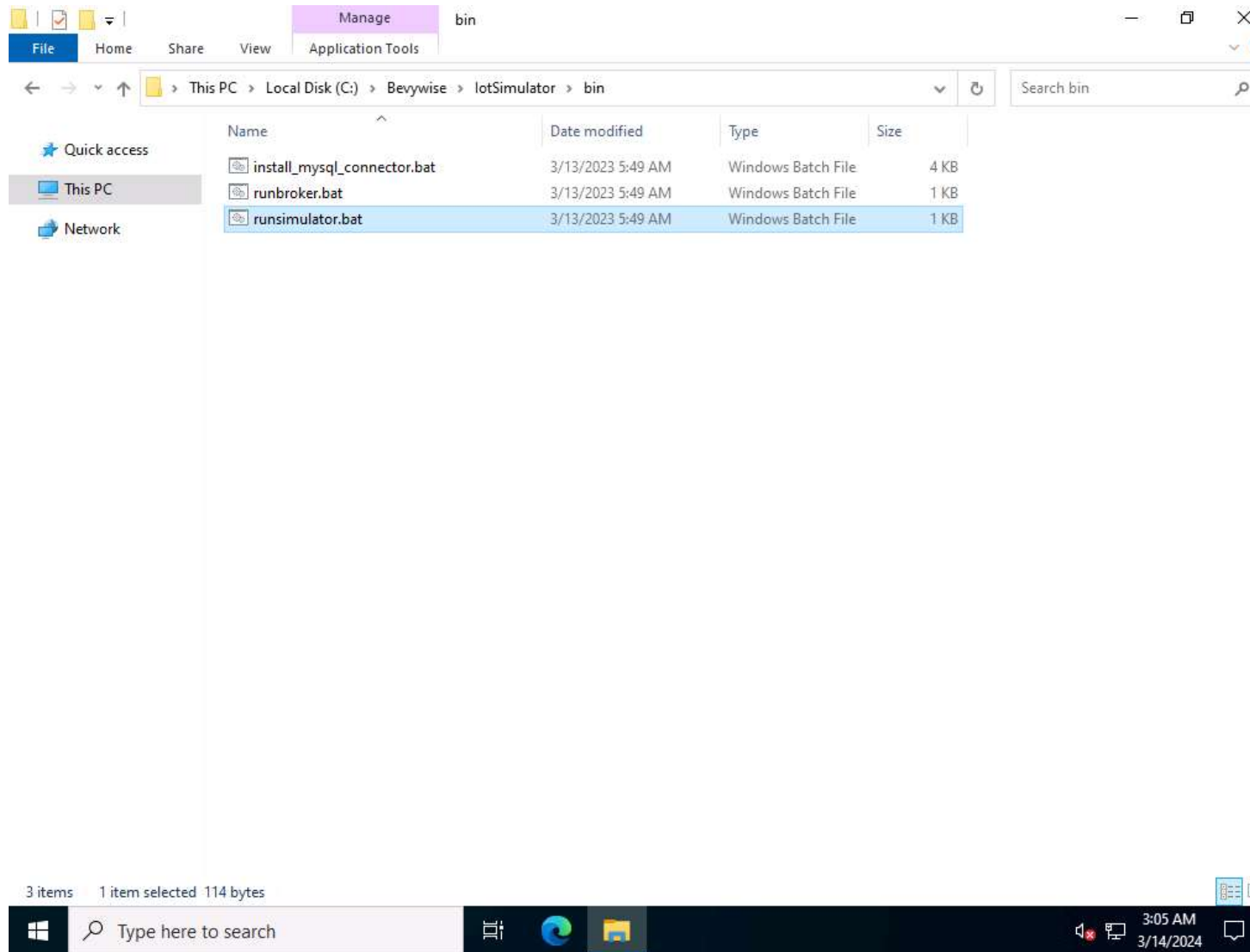
16. If **Open File - Security Warning** popup appears, click **Run..**
17. The **Setup-IoTSimulator\_3 3.0** setup wizard opens. Select **I accept the agreement** and follow the wizard driven steps.
18. To complete the installation, select **Yes, restart the computer now** and click on **Finish** to complete the installation.
19. If restart computer option does not appear, then continue from **Step#16**.

20.



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

22.



23. 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](http://127.0.0.1:9000/setnetwork?network=HEALTH_CARE)**.
24. If the URL directly opens in Microsoft Edge browser, then continue.
25. 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.

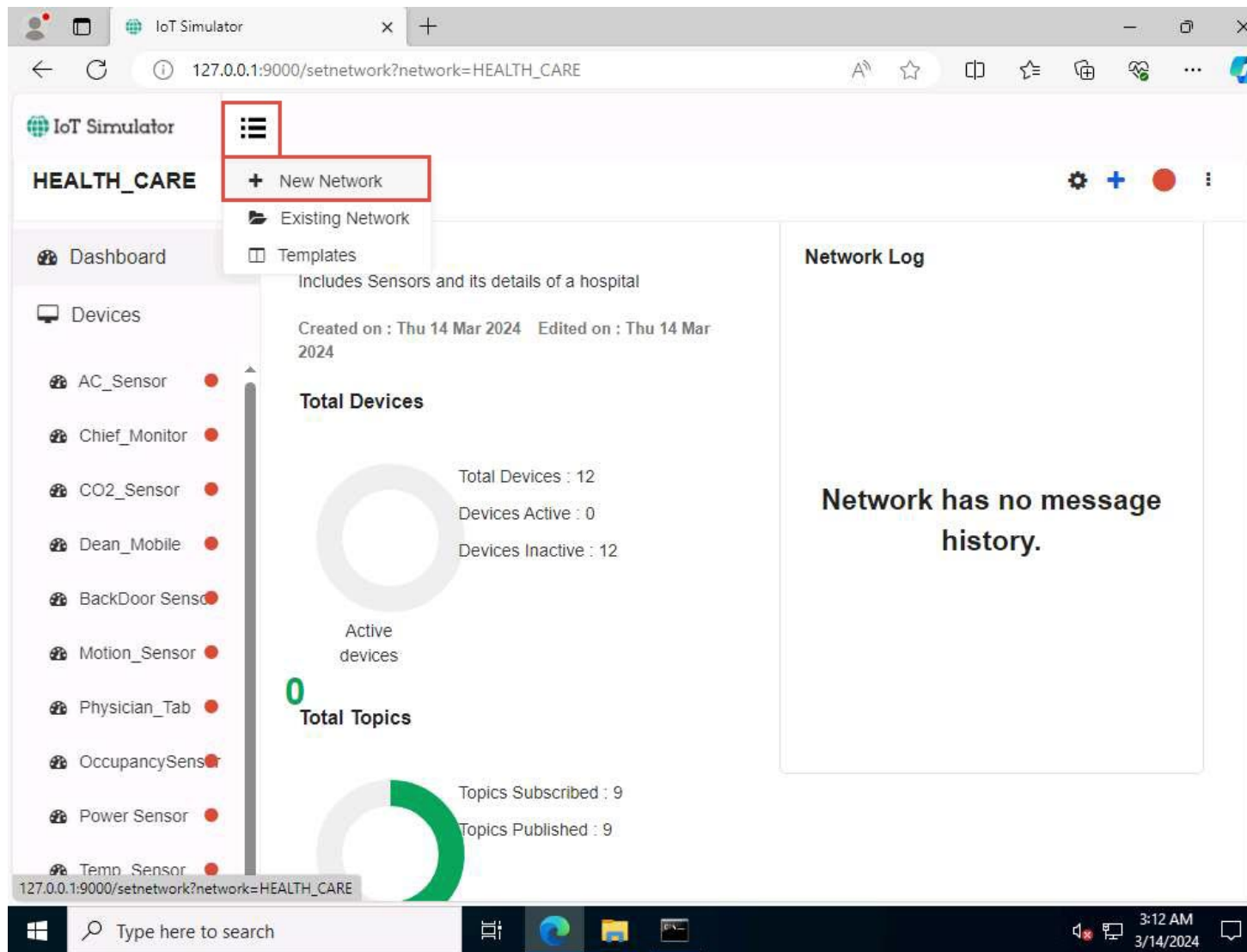
26.

The screenshot displays the IoT Simulator web application. The browser address bar shows the URL `127.0.0.1:9000/setnetwork?network=HEALTH_CARE`. The application header includes the "IoT Simulator" logo and a menu icon. The main content area is titled "HEALTH\_CARE" and features a sidebar with navigation options: "Dashboard", "Devices", and a list of sensors (AC\_Sensor, Chief\_Monitor, CO2\_Sensor, Dean\_Mobile, BackDoor Sensor, Motion\_Sensor, Physician\_Tab, OccupancySens, Power Sensor, Temp\_Sensor). The main panel is divided into three sections: "Description" (stating it includes sensors and hospital details, created and edited on Thu 14 Mar 2024), "Total Devices" (showing 12 total devices, 0 active, and 12 inactive with a donut chart), and "Total Topics" (showing 9 topics subscribed and 9 topics published with a donut chart). A "Network Log" section on the right states "Network has no message history." The Windows taskbar at the bottom shows the search bar, task view, and system clock (3:11 AM, 3/14/2024).

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

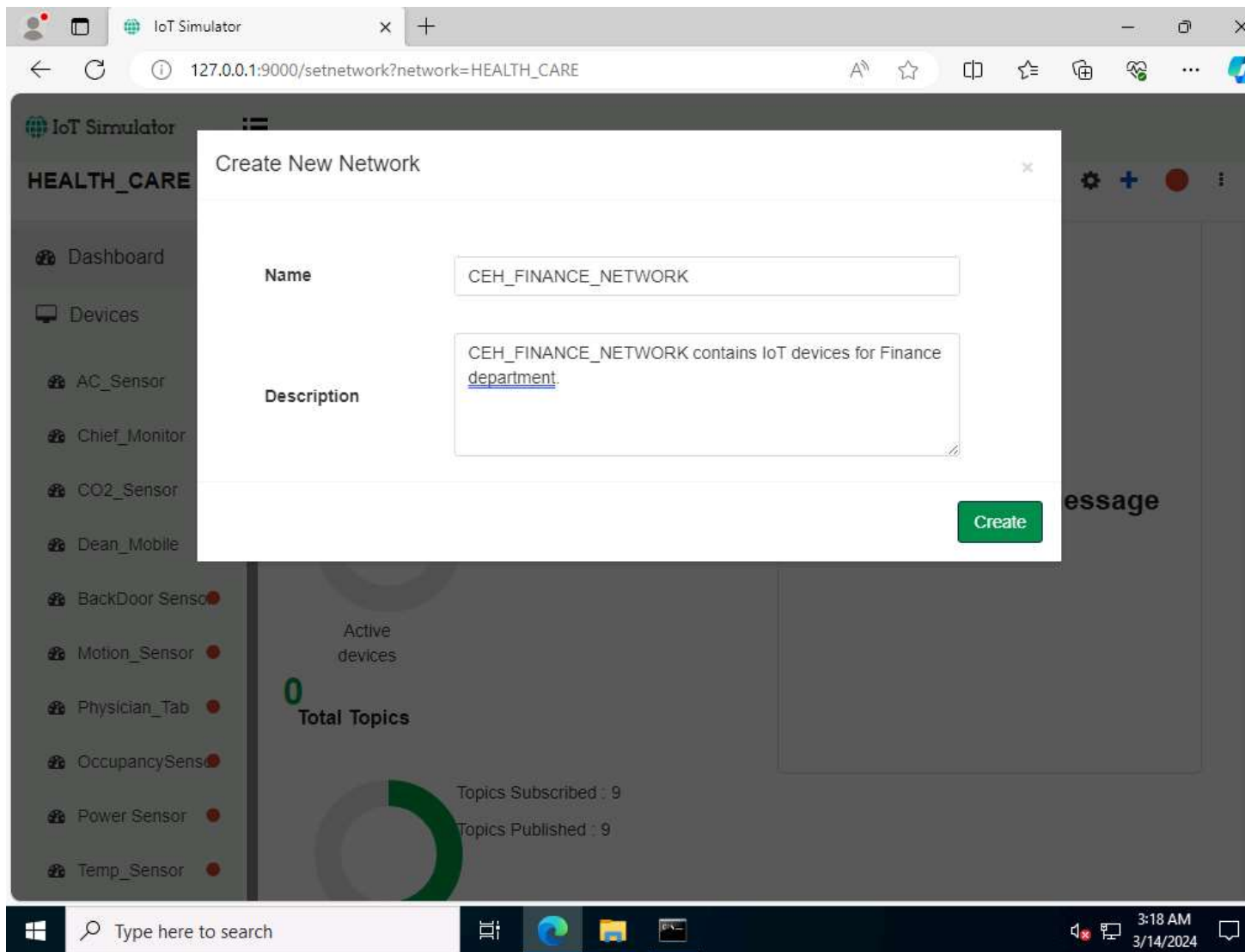


28.



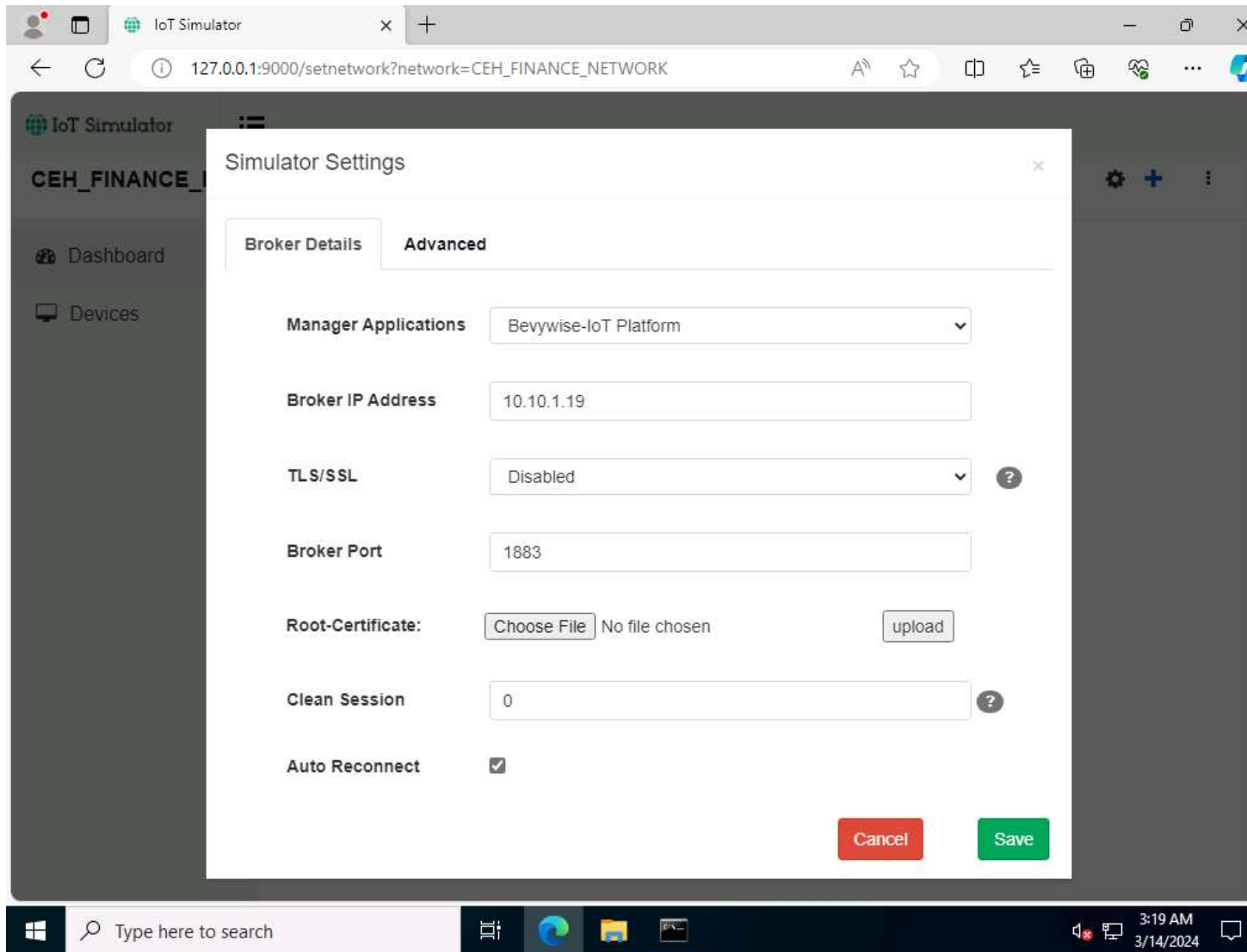
29. The **Create New Network** popup appears. Type any name (here, **CEH\_FINANCE\_NETWORK**) and description. Click on **Create**.

30.



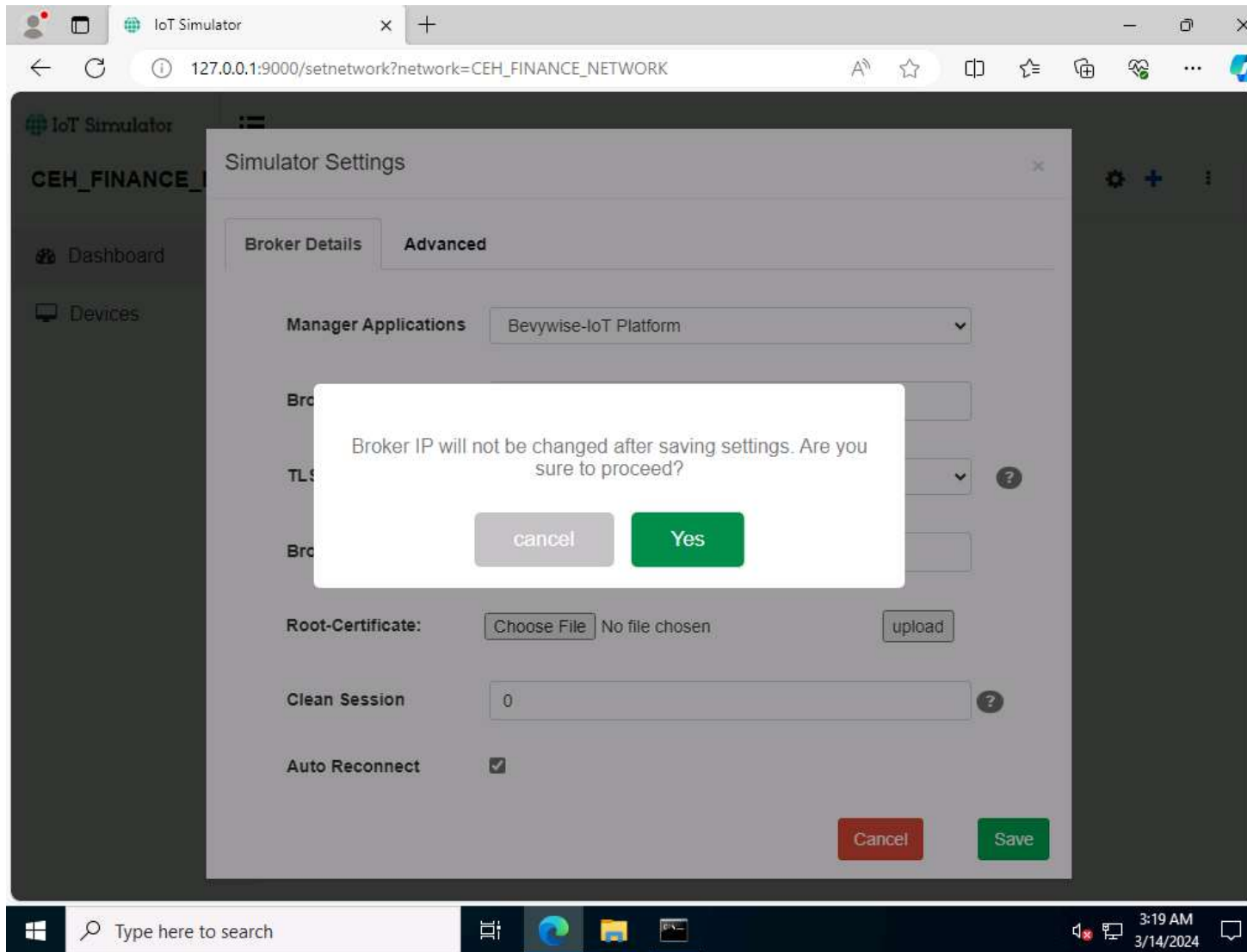
31. In the next screen, we will setup the **Simulator Settings**. Set the **Broker IP Address** as **10.10.1.19** (the IP address of the **Windows Server 2019** ). Since we have installed the Broker on the web server, the created network will interact with the server using MQTT Broker. Do not change default settings and click on **Save**.

32.



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

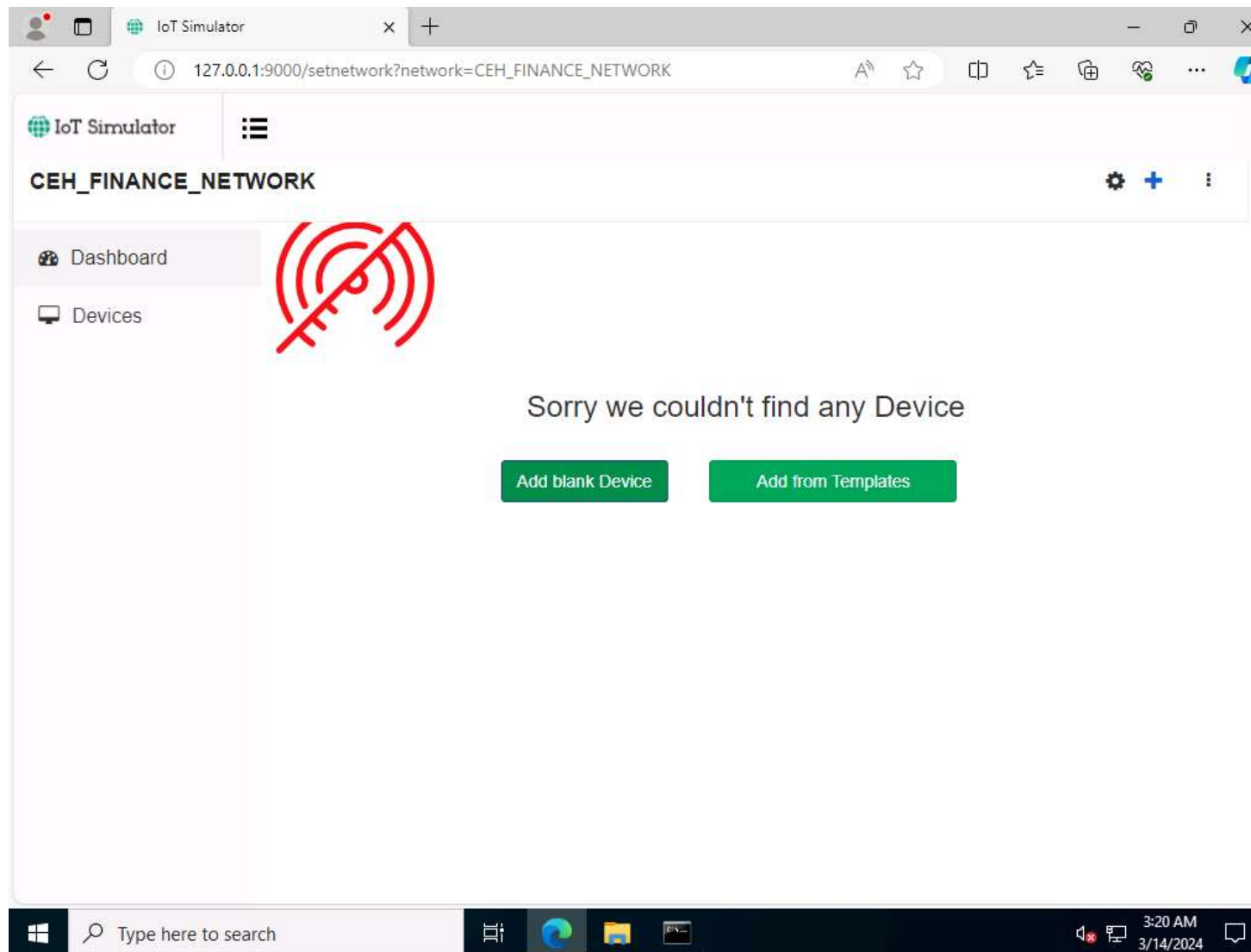
34.



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

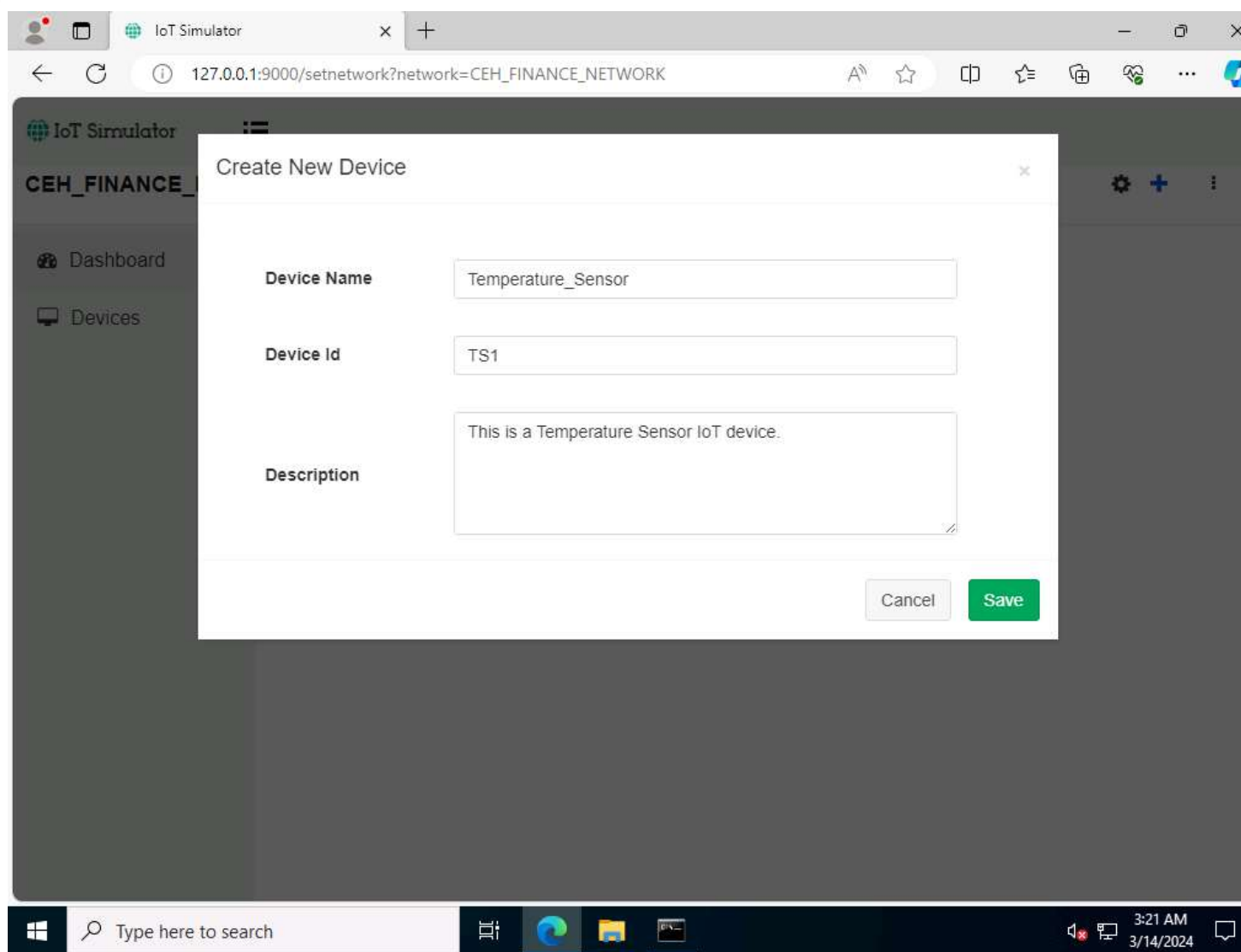
36. To add IoT devices to the created network, click on the **Add blank Device** button.

37.



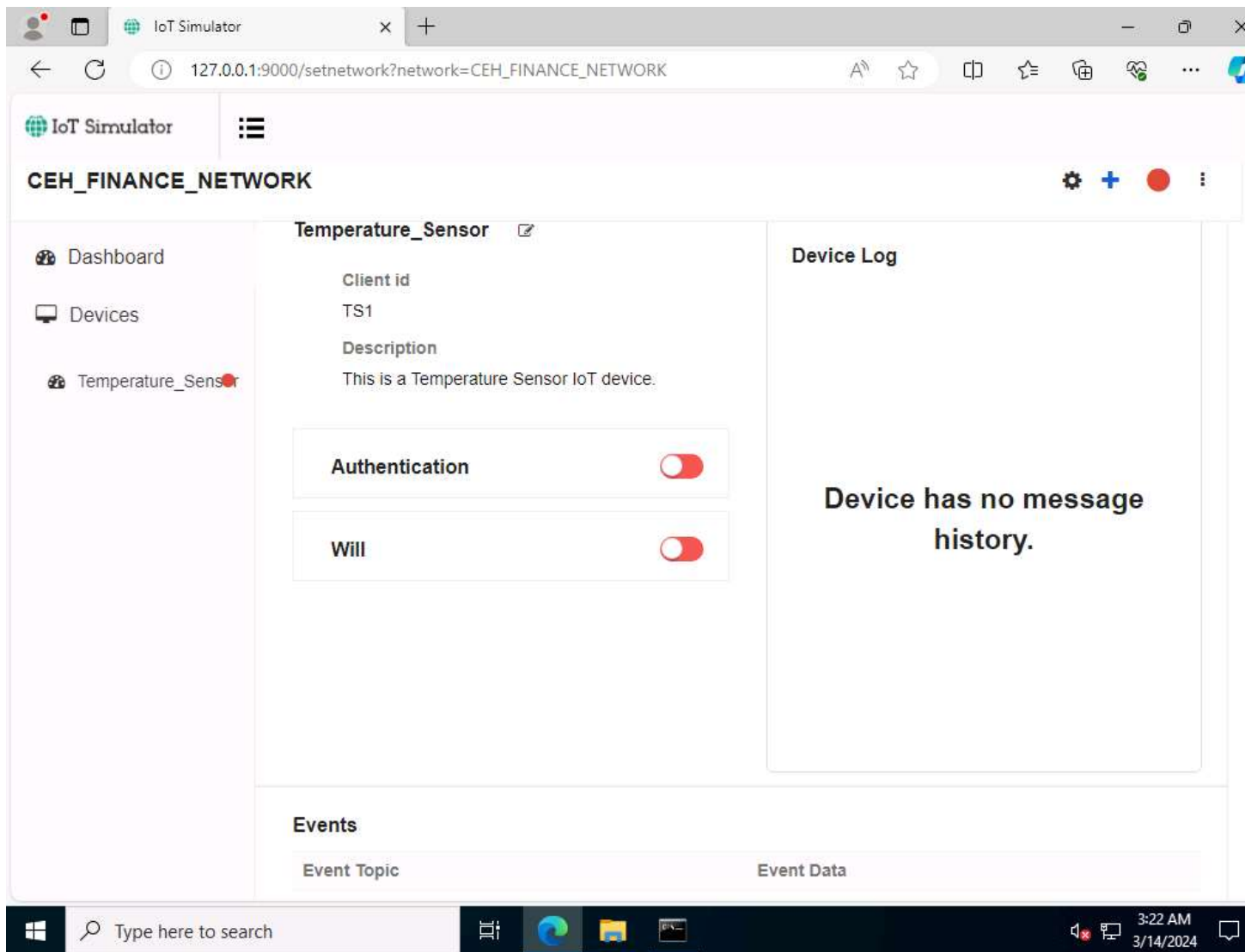
38. The **Create New Device** popup opens. Type the device name (here, we use **Temperature\_Sensor**), enter Device Id (here, we use **TS1**), provide a **Description** and click on **Save**.

39.



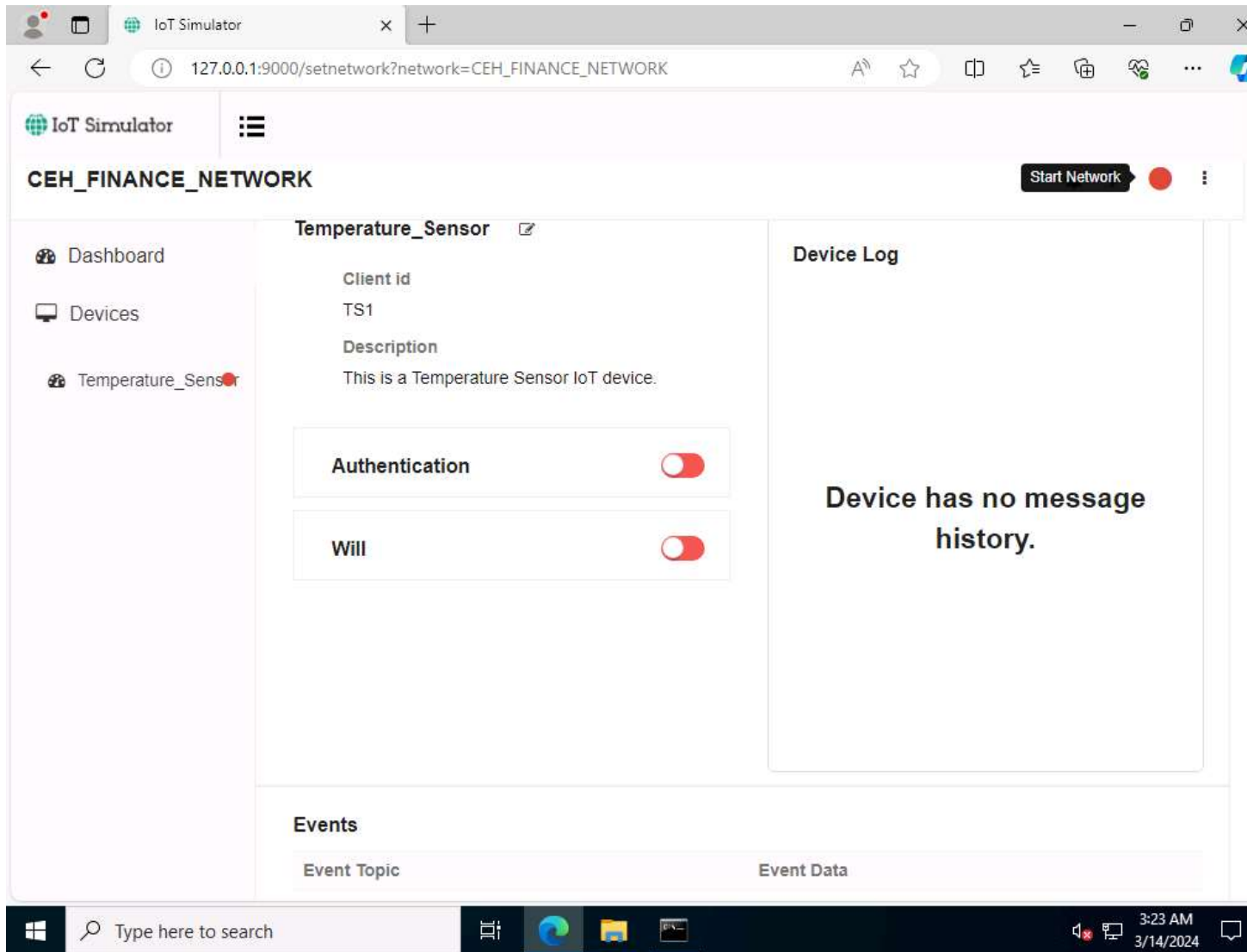
40. The device will be added to the **CEH\_FINANCE\_NETWORK**.

41.



42. 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.

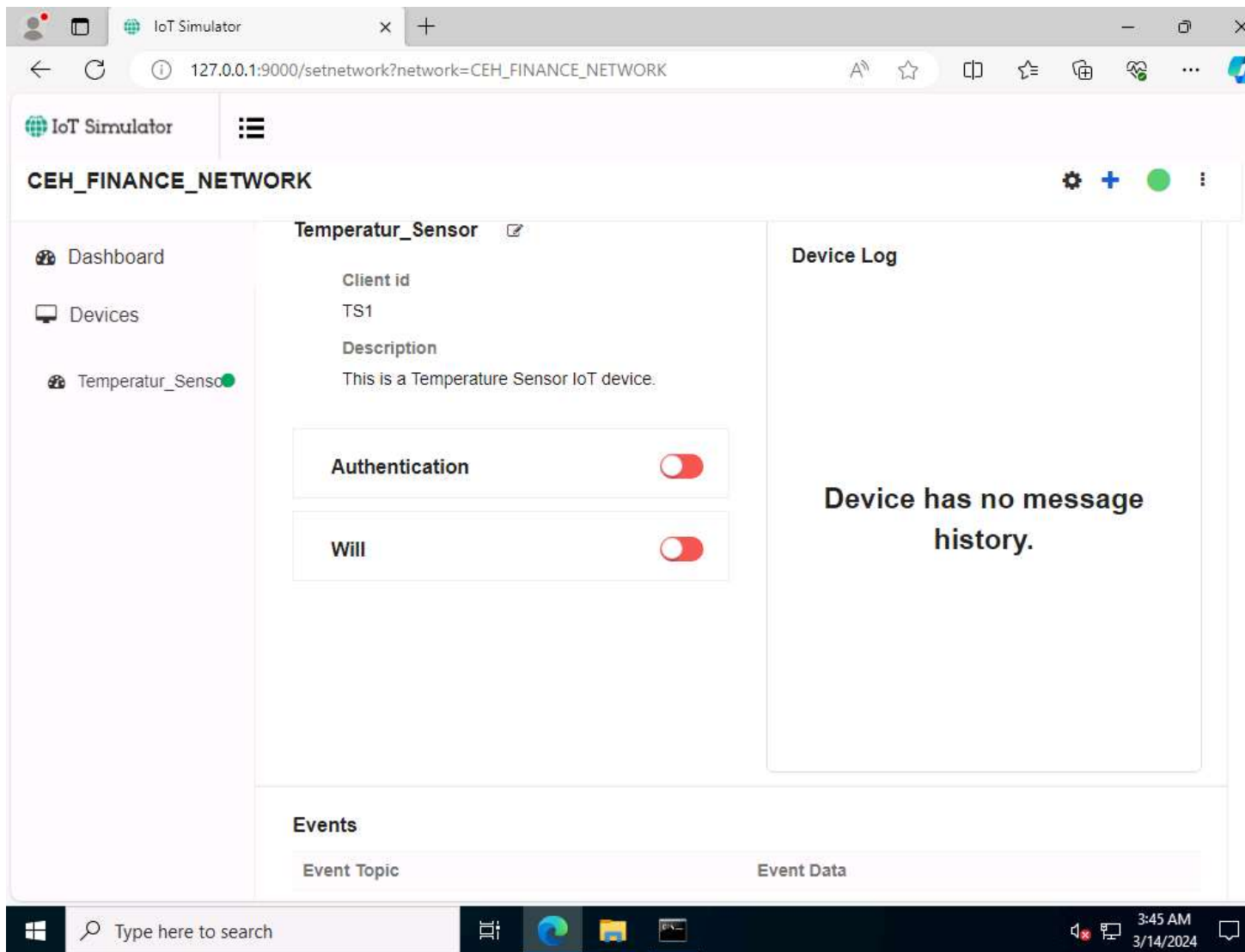
43.



44. 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**.

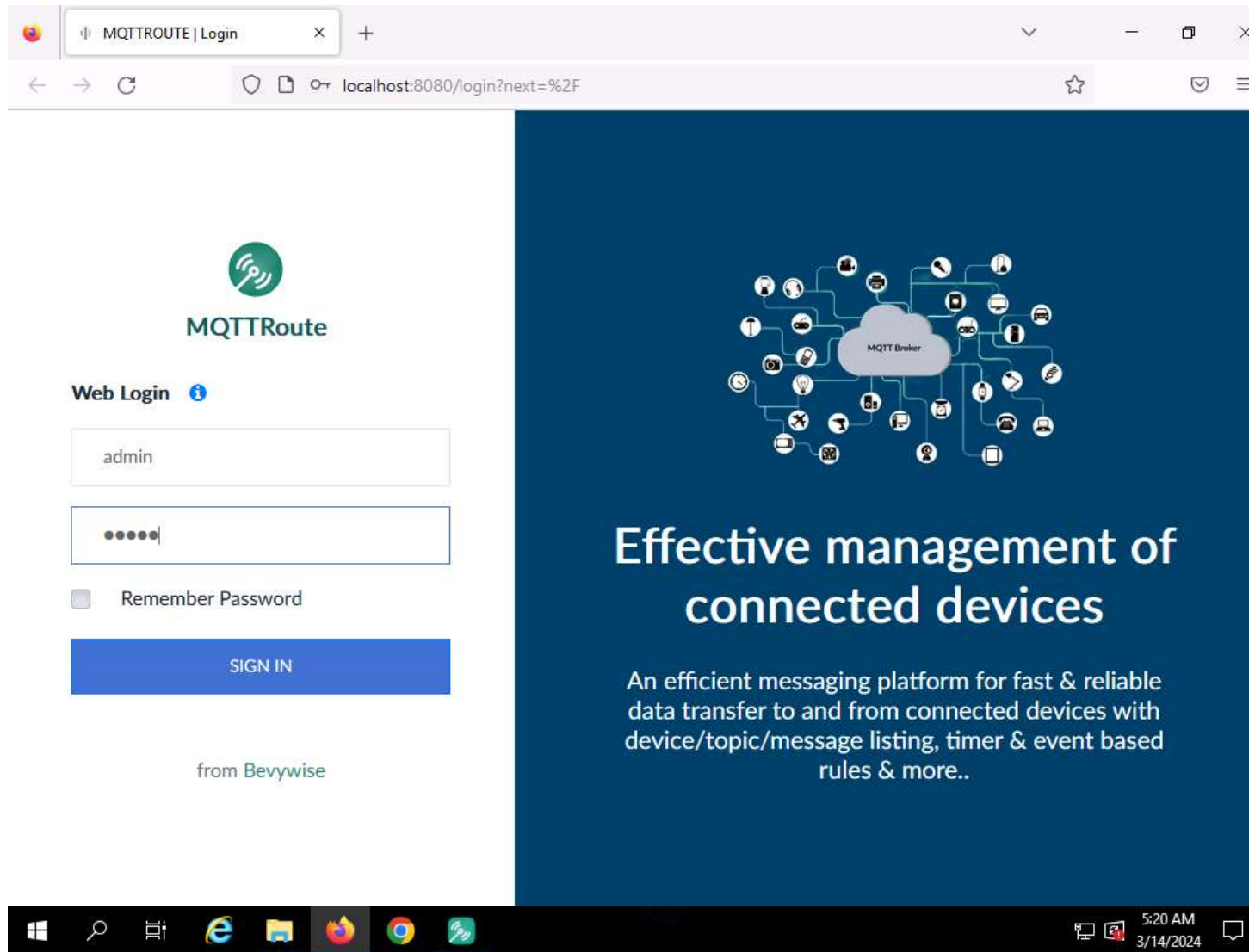


45.



46. 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).

47.



48. 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.

49.

The screenshot shows the MQTTRoute web interface in a browser window. The address bar indicates the URL is localhost:8080. The interface features a top navigation bar with a 'MQTTRoute' tab and a plus icon for additional tabs. Below the navigation bar, there are four summary cards: 'Active Devices' (1), 'Total Devices' (1), 'Events' (0), and 'Commands' (0). The main content area is divided into three sections: 'Recent Events', 'Recent Device Log', and 'Recent Connections'. The 'Recent Events' and 'Recent Device Log' sections are currently empty, displaying 'No Data Found'. The 'Recent Connections' section contains a single entry for device 'TS1' at IP '10.10.1.22' with a connection time of 'Today 05:15:32'. A red rectangle highlights this entry. The 'Recent Disconnections' section is also empty, displaying 'No Data Found'. The Windows taskbar at the bottom shows the system clock as 5:21 AM on 3/14/2024.

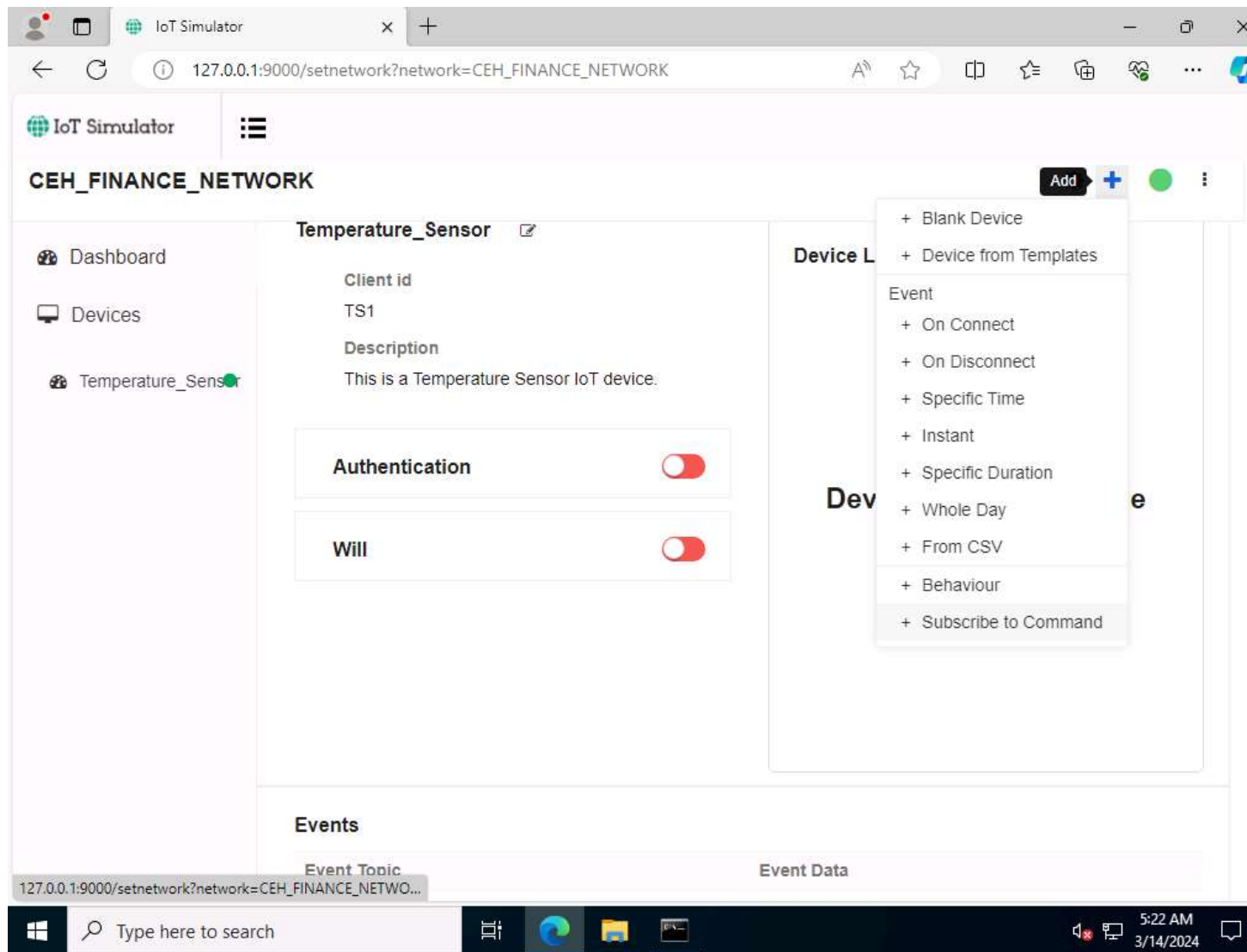
Device Id	IP	Time
TS1	10.10.1.22	Today 05:15:32

50. Switch back to [Windows Server 2022](#) machine.

51. Next, we will create the **Subscribe command** for the device Temperature\_Sensor.

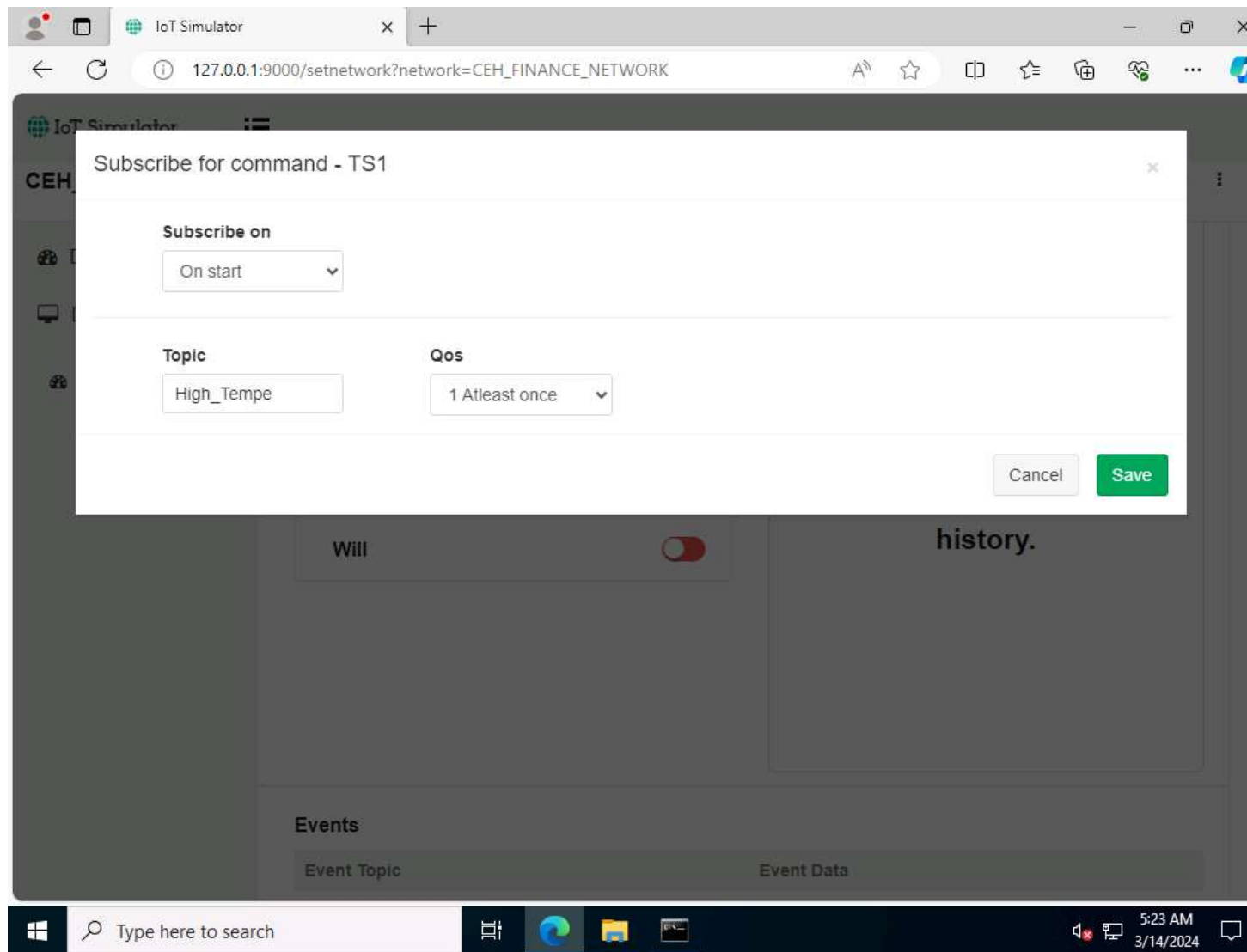
52. Click on the **Plus** icon in the **top right corner** and select the **Subscribe to Command** option.

53.



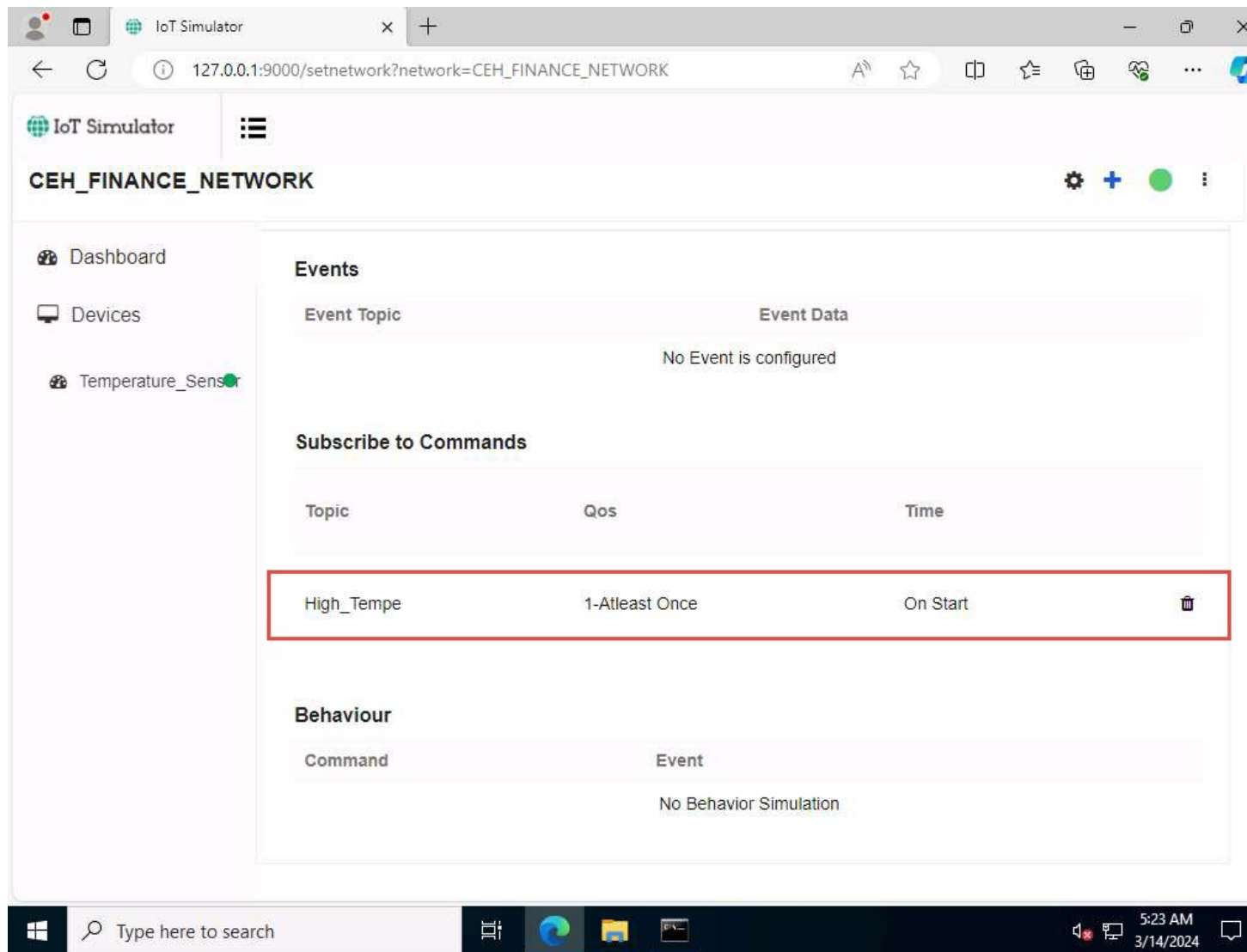
54. The **Subscribe for command - TS1** popup opens. Select **On start** under the **Subscribe on** tab, type **High\_Tempe** under the **Topic** tab, and select **1 Atleast once** below the **Qos** option. Click on **Save**.

55.



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

57.



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

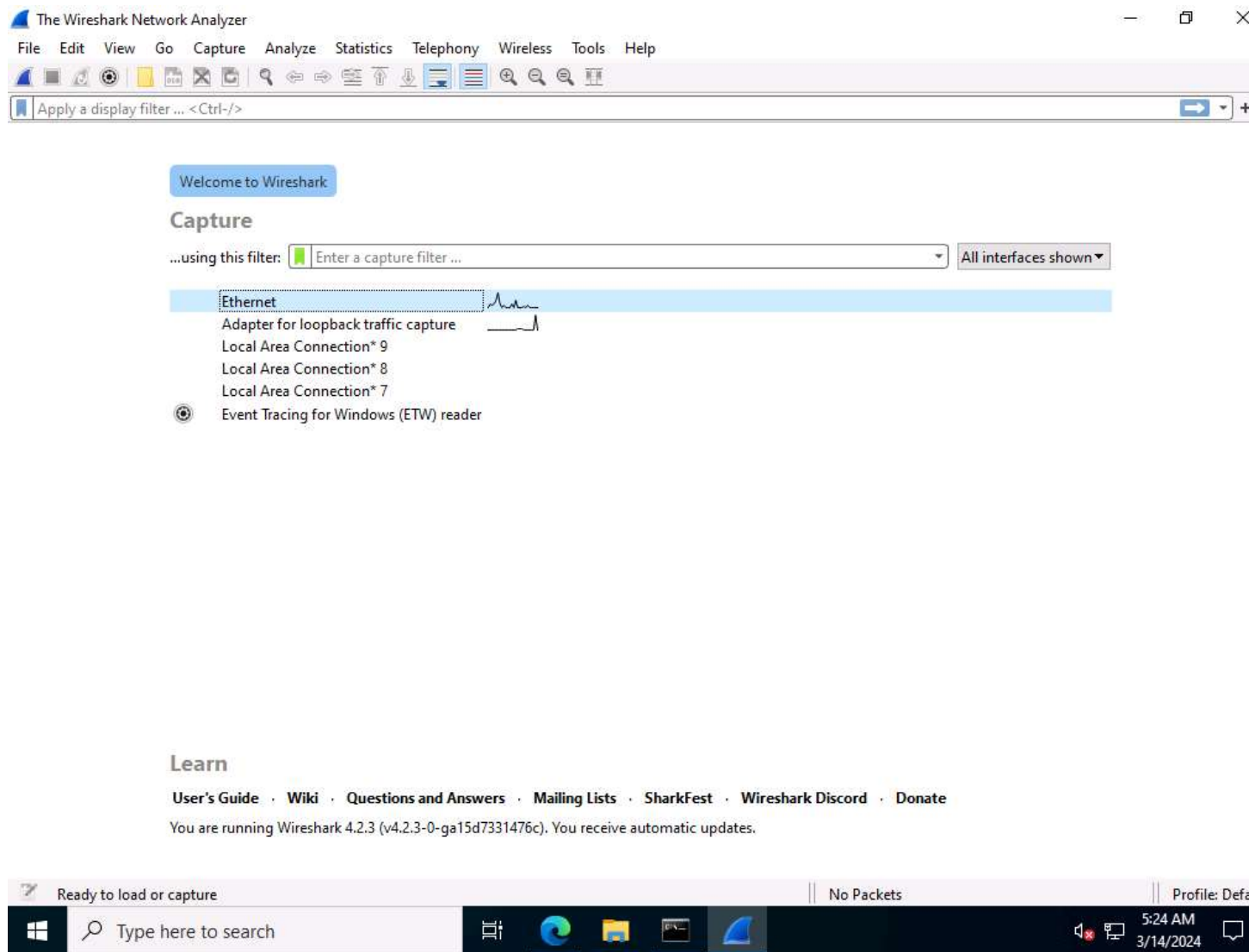
59. Minimise the Edge browser. Click **Type here to search** field on the **Desktop**, search for **wireshark** in the search bar and select **Wireshark** from the results.

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

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

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

63.



64. Click on the **Start Wireshark** icon to start the capturing packets, leave the Wireshark running.
65. Leave the IoT simulator running and switch to the [Windows Server 2019](#) machine.
66. Navigate to **Devices** menu and click on connected device i.e.**TS1**.

67.

The screenshot shows a web browser window with the URL `localhost:8080/#page-single-device`. The page title is "MQTTRoute". The navigation bar includes "Dashboard", "Devices" (highlighted with a red box), "Topics", "Rules", and "Device Log". Below the navigation bar is a table with columns: "Device Name", "Device Id", "Status", "Will Topic", "Will Qos", "Will Message", and "Time". The table contains one row with "TS1" in the "Device Name" column (highlighted with a red box), "TS1" in the "Device Id" column, and "Active" in the "Status" column. Below the table is a section with tabs: "Events", "Commands", "Subscribe Topics", and "Send Command" (highlighted with a yellow box). The "Send Command" section has a "Topic" dropdown menu with "High\_Tempe" selected, a "Message" text area, and a "Submit" button.

68. Now, we will send the command to **TS1** using the **High\_Tempe** topic.

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



70.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/#page-single-device'. The page title is 'MQTTRoute'. The navigation bar includes links for 'Dashboard', 'Devices', 'Topics', 'Rules', and 'Device Log'. The main content area displays a table with columns: 'Device Name', 'Device Id', 'Status', 'Will Topic', 'Will Qos', 'Will Message', and 'Time'. A single row is visible for device 'TS1', which has a status of 'Active' and a time of 'Today 22:02:28'. Below the table, there are tabs for 'Events', 'Commands', 'Subscribe Topics', and 'Send Command'. The 'Send Command' tab is selected, showing a 'Topic' dropdown set to 'High\_Tempe' and a 'Message' text area containing 'Alert for High Temperature.'. A 'Submit' button is located at the bottom right of the form.

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

Events Commands Subscribe Topics Send Command

Topic: High\_Tempe

Message: Alert for High Temperature.

Submit

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

72.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/#page-single-device'. The page header includes the Bevywise logo and navigation links: Dashboard, Devices, Topics, Rules, and Device Log. Below the header is a table with columns: Device Name, Device Id, Status, Will Topic, Will Qos, Will Message, and Time. A single row is visible for device 'TS1', which is 'Active' and has a timestamp of 'Today 22:02:28'. Below the table, there are four tabs: Events, Commands, Subscribe Topics, and Send Command. The 'Send Command' tab is selected. It contains a 'Topic' dropdown menu with 'High\_Tempe' selected, a 'Message' text input area, and a 'Submit' button. A green message box at the bottom left of the form area says 'Message send to TS1'.

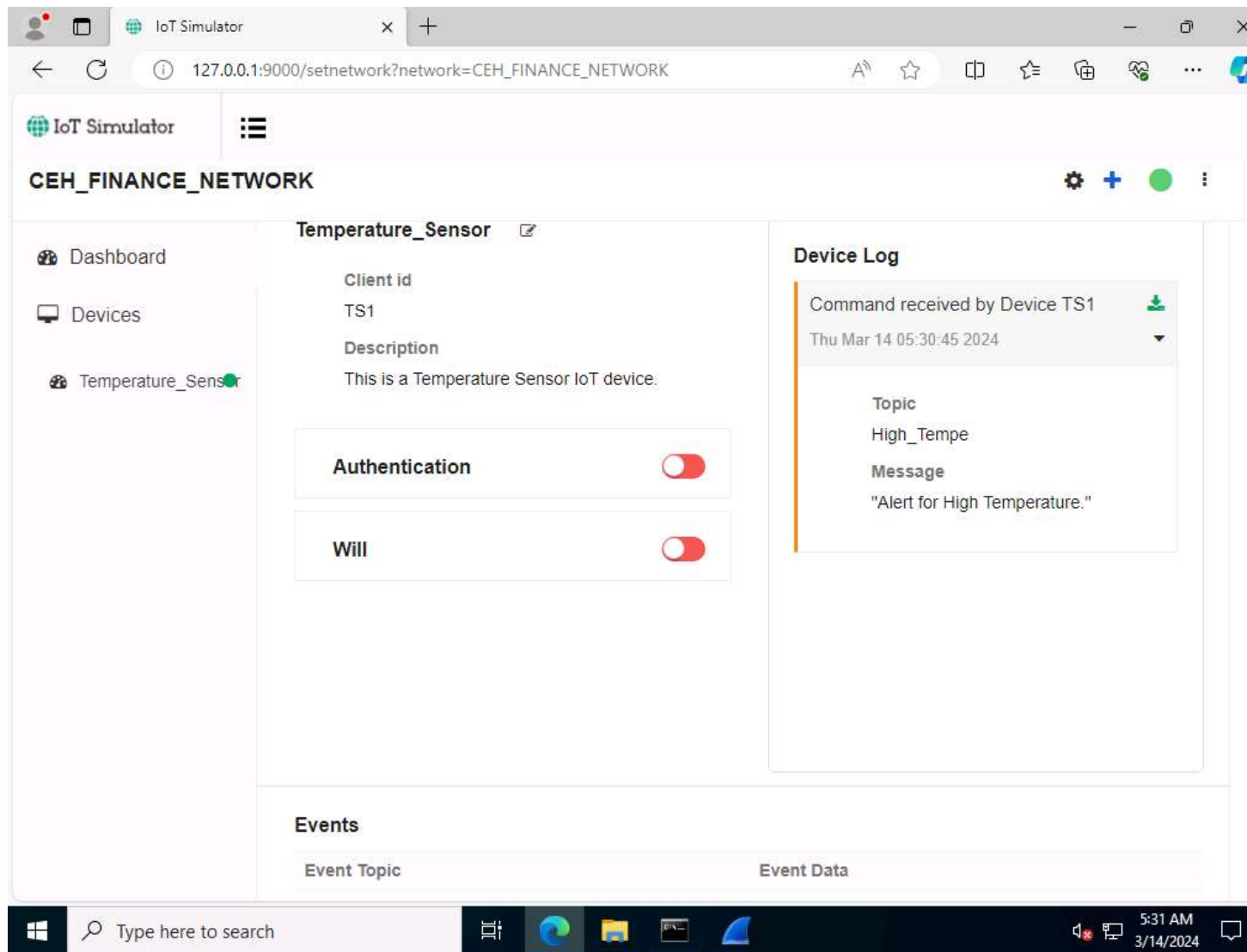
73. The message has been sent to the device using this topic.

74. Next, switch to [Windows Server 2022](#) machine.

75. 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.

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

77.



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

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

The figure displays a Wireshark network traffic analysis window titled "Ethernet". The main pane shows a list of captured packets, all belonging to the MQTT protocol. The packets are organized into three groups based on their source and destination IP addresses:

- Group 1 (Source: 10.10.1.22, Destination: 10.10.1.19):** Contains Ping Request and Ping Response messages.
- Group 2 (Source: 10.10.1.19, Destination: 10.10.1.22):** Contains Ping Request and Ping Response messages.
- Group 3 (Source: 10.10.1.22, Destination: 10.10.1.22):** Contains Publish Message, Publish Ack, Publish Received, Publish Release, and Publish Complete messages, all associated with ID=2 [High\_Tempe].

The bottom pane provides detailed information for the selected packet (Frame 39), which is a "MQ Telemetry Transport Protocol, Ping Request". It specifies the frame size (56 bytes on wire/captured) and lists the involved protocols: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and MQ Telemetry Transport Protocol.

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

Detailed view of Frame 39:

```
> 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 (00:15:5d:01:80:02)
> 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
```

81. 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.
82. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Topic Length**, **Topic**, and **Message**.
83. Publish Message can be used to obtain the message sent by the MQTT client to the broker.

84.

The image shows a Wireshark capture of MQTT traffic on the 'mqtt' interface. The packet list shows a sequence of PING requests and responses, followed by a Publish Message (id=2) on the topic 'High\_Tempe'. The packet details pane for the selected Publish Message shows the following structure:

- Frame 679: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface \Device\NPF\_{6F8...}
- Ethernet II, Src: MS-NLB-PhysServer-21\_5d:35:38:46 (02:15:5d:35:38:46), Dst: Microsoft\_01:80:02 (08:00:02:01:80:02)
- 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
- 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 (Acknowledged)
  - Msg Len: 41
  - Topic Length: 10
  - Topic: High\_Tempe
  - Message Identifier: 2
  - Message: 416c65727420666f7220486967682054656d70657261747572652e

The packet bytes pane shows the raw data of the message, which is a UTF-8 string: "High Temperature".

85. 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:

- o Header Flags: Contains information regarding the MQTT control packet type.
- o 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.
- o QoS: Determines the assurance level of a message.
- o 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.
- o Topic Name: Contains a UTF-8 string that can also include forward slashes when it needs to be hierarchically structured.
- o Message: Contains the actual data to be transmitted.
- o Payload: Contains the message that is being published.

86. 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.

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



88.

The screenshot shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for packet capture and analysis. The packet list pane (top) shows a list of captured packets, with the selected packet being an MQTT 'Publish Release' packet (No. 683, Time 343.087487, Source 10.10.1.19, Destination 10.10.1.22, Protocol MQTT, Length 58, Info 58 Publish Release (id=2)). The packet details pane (middle) is expanded to show the 'MQ Telemetry Transport Protocol, Publish Release' section, displaying fields such as 'Msg Len: 2' and 'Message Identifier: 2'. The packet bytes pane (bottom) shows the raw data of the packet in hexadecimal and ASCII.

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

90. 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.

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

92.

The image shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for packet capture and analysis. The packet list pane at the top shows a list of captured packets, with the following details visible:

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

Packet 684 is selected, and its details are shown in the packet details pane. The details pane shows the structure of the MQTT message, including the following sections:

- 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]
- 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

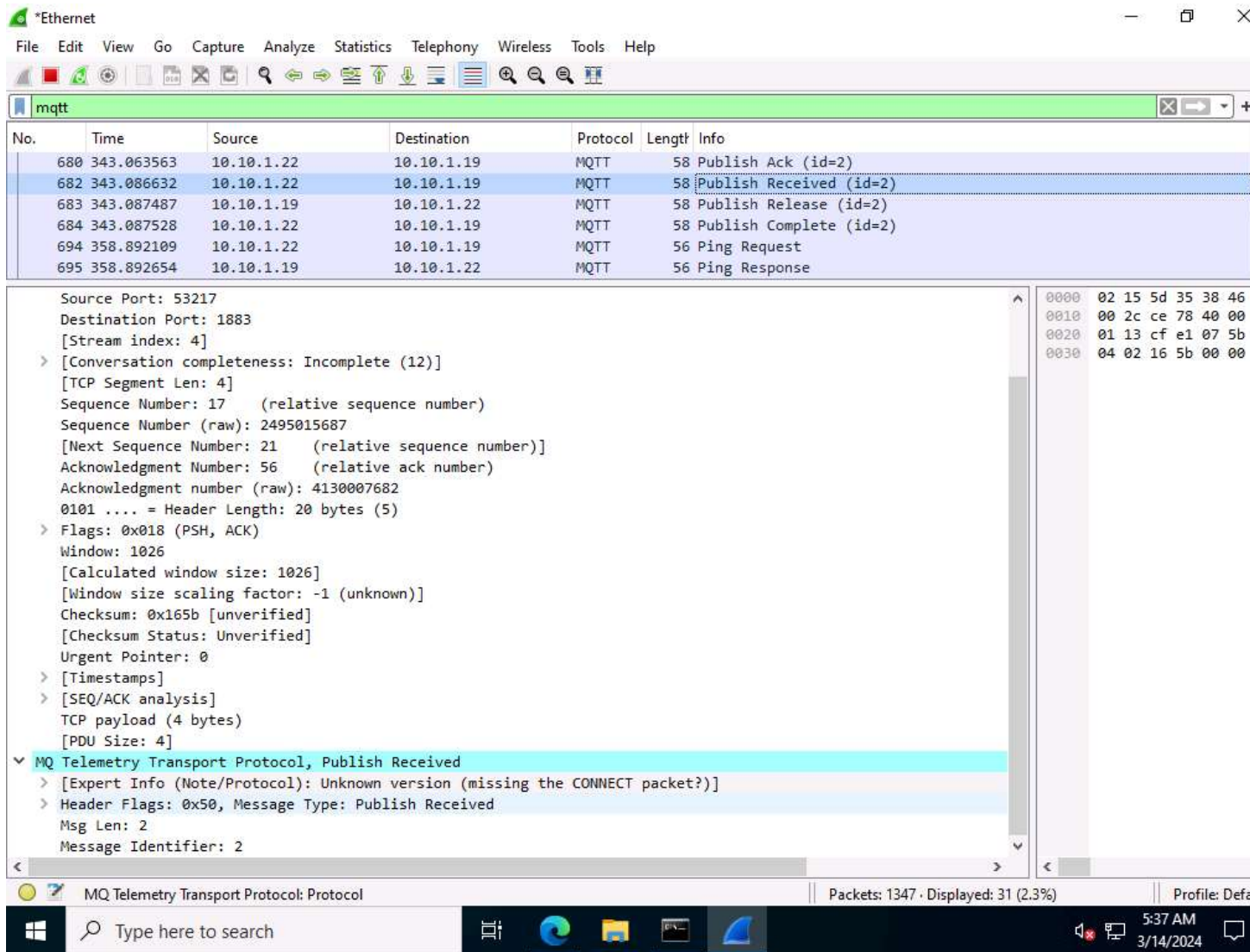
The bottom status bar shows "Packets: 1294 · Displayed: 31 (2.4%)" and "Profile: Defa".

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

94. 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.

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

96.



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

98. 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.

99. 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?

Score

#### 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?

Score

### 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.

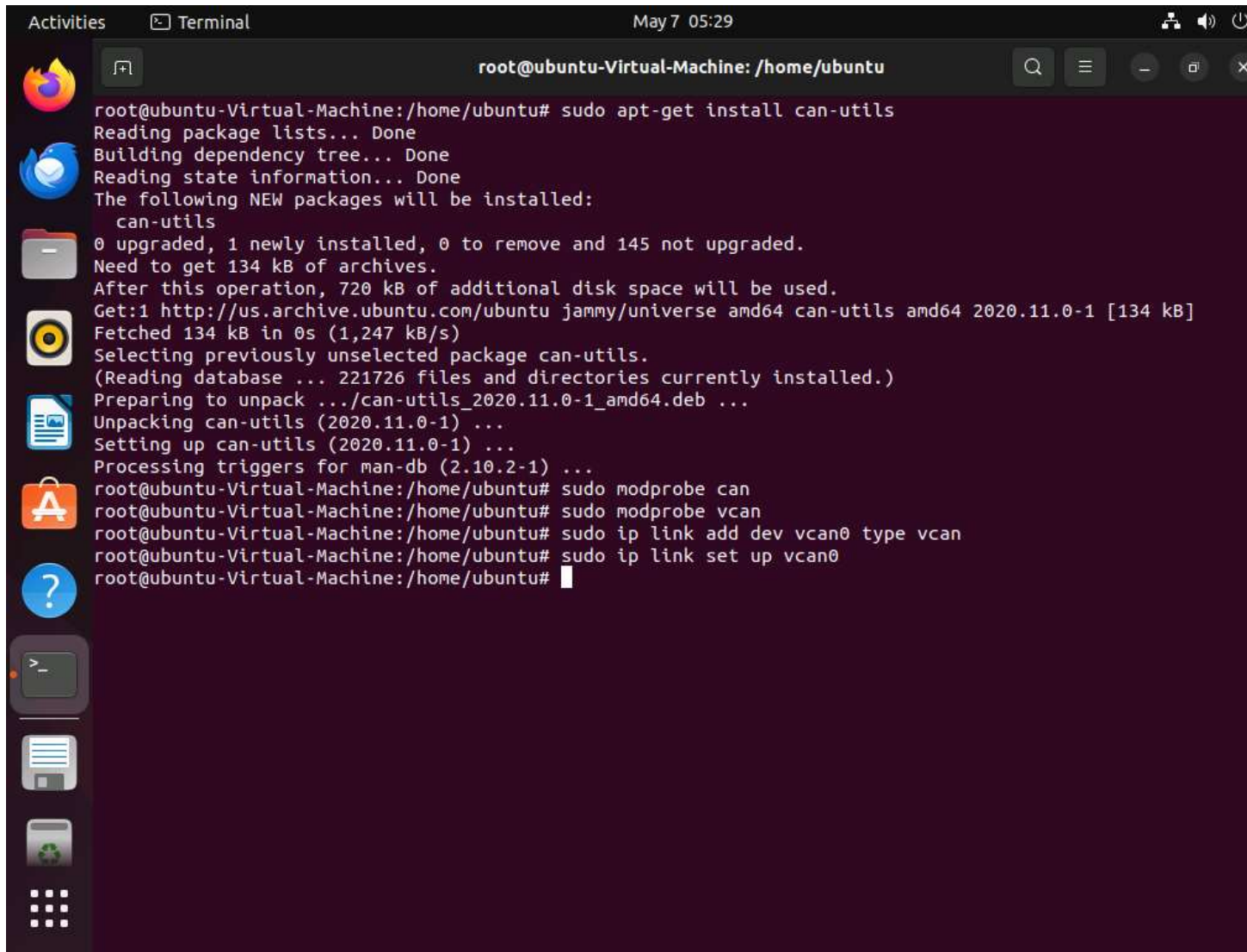
## Task 1: Perform Replay Attack on CAN Protocol

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. The **can-utils** package is already installed on the system.
4. Now, to setup a virtual CAN interface issue following commands:
  - o **sudo modprobe can**
  - o **sudo modprobe vcan**
  - o **sudo ip link add dev vcan0 type vcan**
  - o **sudo ip link set up vcan0**

5.

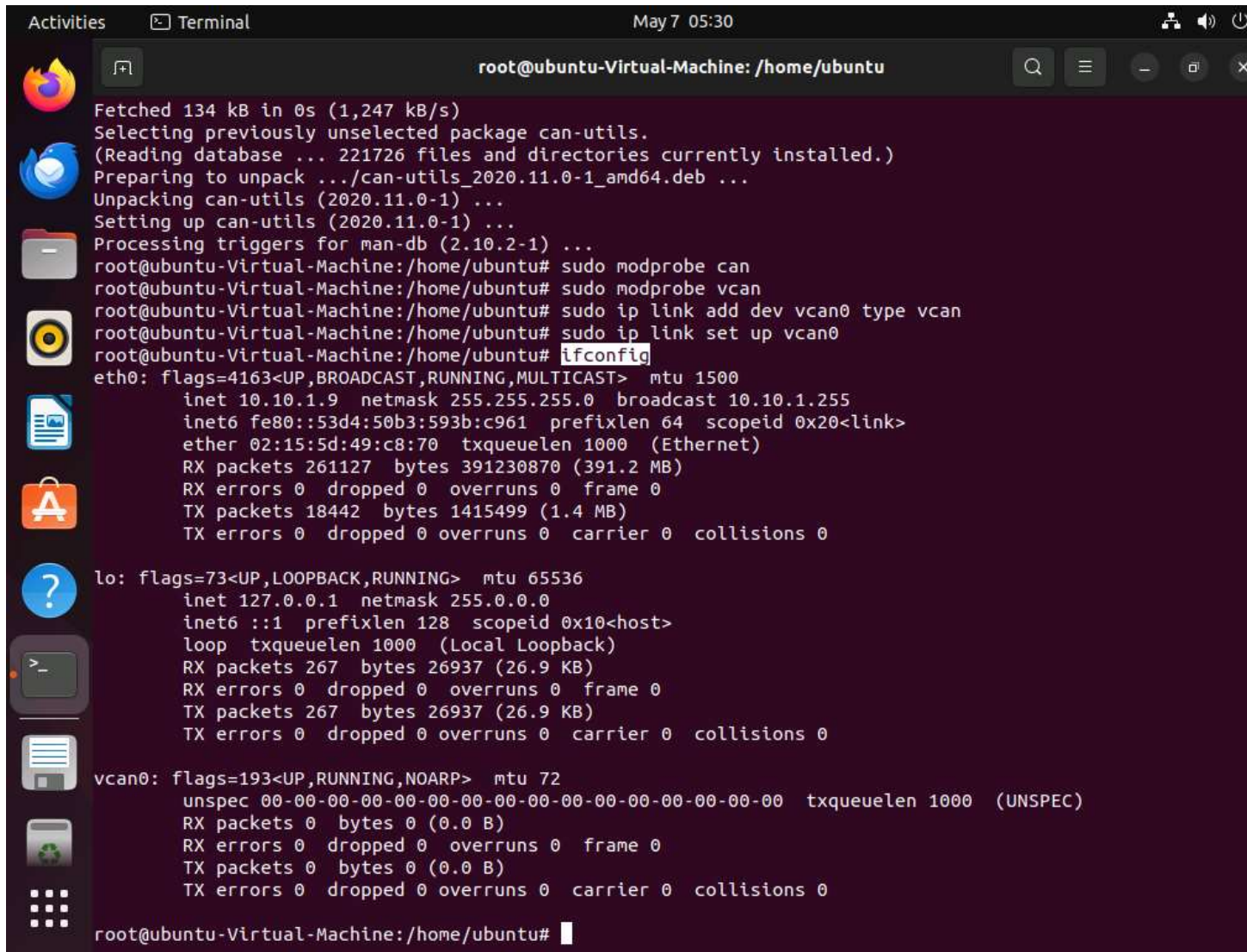


```
Activities Terminal May 7 05:29
root@ubuntu-Virtual-Machine: /home/ubuntu

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#
```

6. 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.

7.



```
Activities Terminal May 7 05:30
root@ubuntu-Virtual-Machine: /home/ubuntu

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

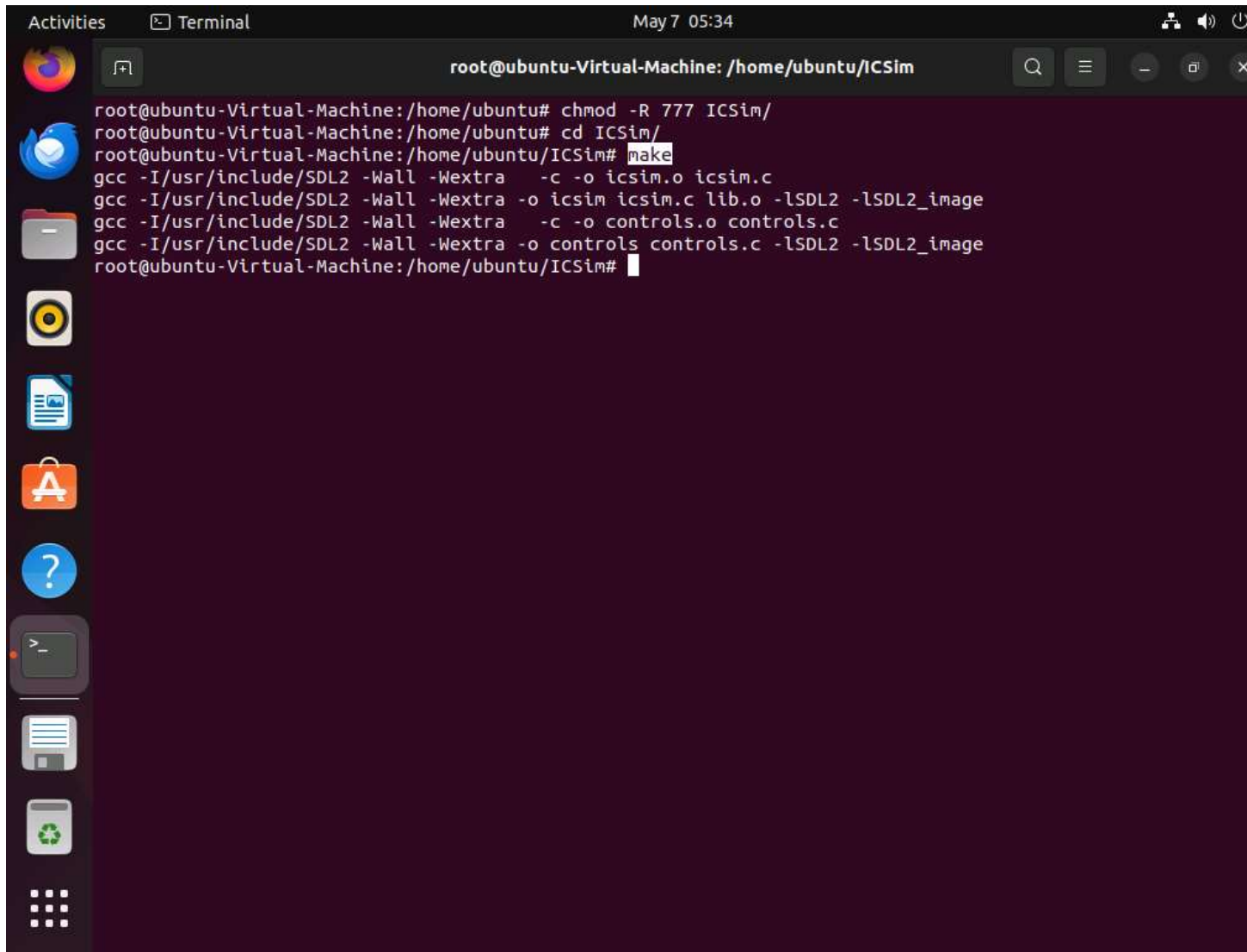
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 267 bytes 26937 (26.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 267 bytes 26937 (26.9 KB)
   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

8. Run **chmod -R 777 ICSim** to give permissions to the ICSim folder.
9. 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.

10.



The screenshot shows a terminal window titled "Terminal" with the date and time "May 7 05:34". The terminal is running on a system named "root@ubuntu-Virtual-Machine" with the current directory set to "/home/ubuntu/ICSim". The user has executed the following commands:

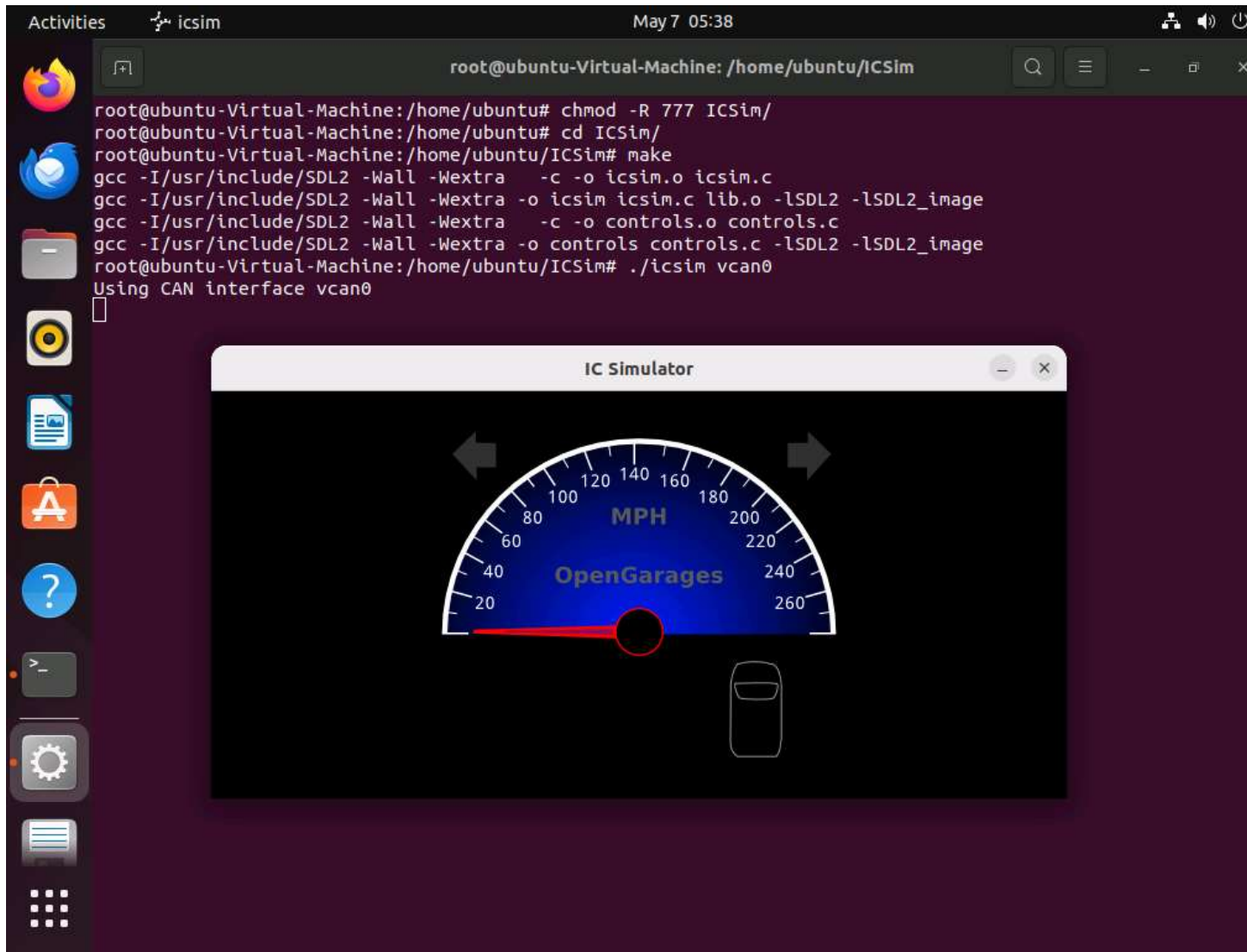
```
root@ubuntu-Virtual-Machine:/home/ubuntu# chmod -R 777 ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu# cd ICSim/
root@ubuntu-Virtual-Machine:/home/ubuntu/ICSim# make
gcc -I/usr/include/SDL2 -Wall -Wextra -c -o icsim.o icsim.c
gcc -I/usr/include/SDL2 -Wall -Wextra -o icsim icsim.c lib.o -lSDL2 -lSDL2_image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

The terminal window is part of a desktop environment with a sidebar on the left containing various application icons, including a web browser, a file manager, and a terminal icon. The terminal window has a title bar with standard window controls (minimize, maximize, close) and a search icon.

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



12.



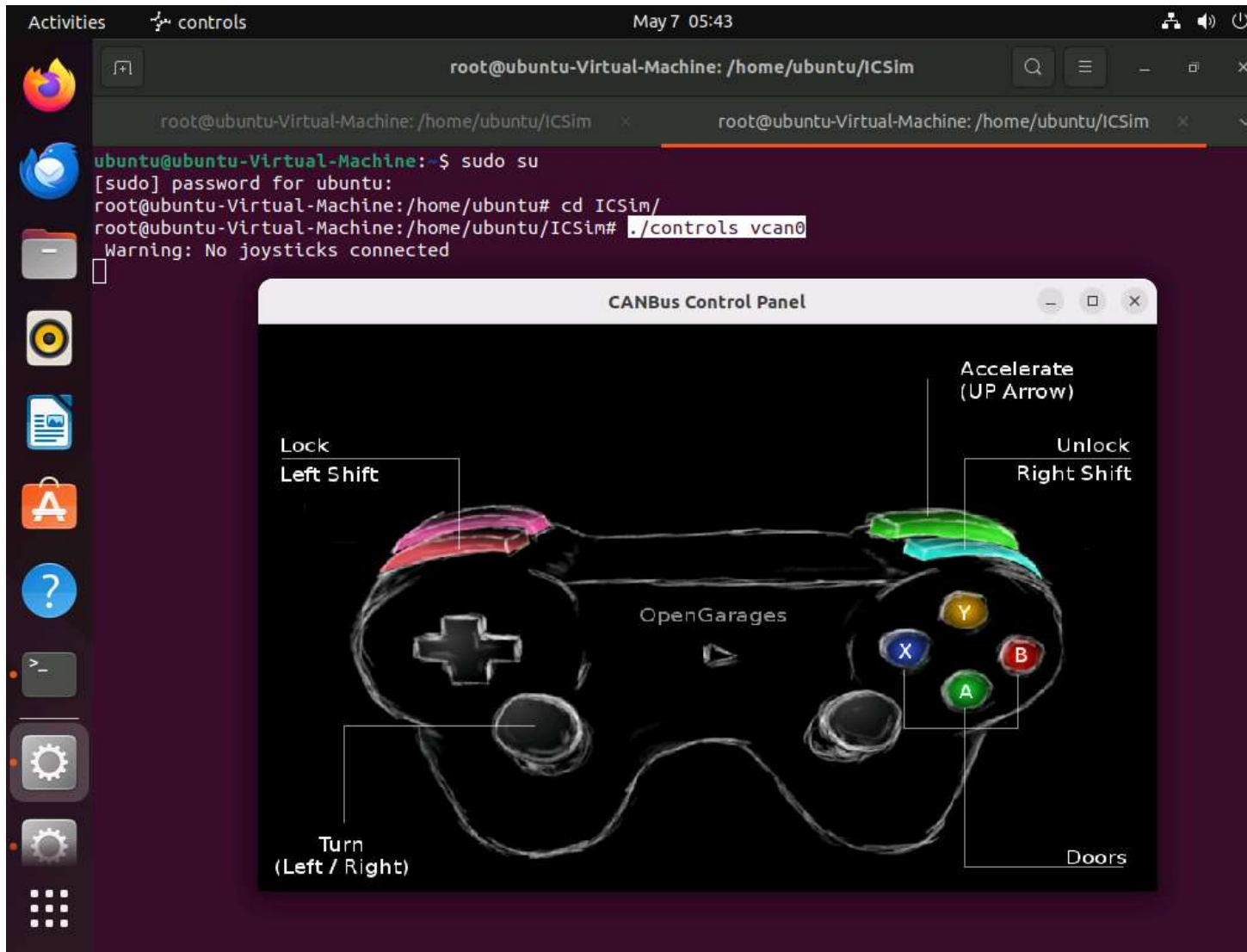
The screenshot shows a Linux desktop environment with a terminal window and an application window titled "IC Simulator". The terminal window has a title bar that says "root@ubuntu-Virtual-Machine: /home/ubuntu/ICSim". The terminal output shows the following commands and their results:

```
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
█
```

The "IC Simulator" window displays a speedometer with a scale from 0 to 260 MPH. The needle is positioned at 0. The text "OpenGarages" is displayed in the center of the speedometer. Below the speedometer is a small icon of a car.

13. 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/**.
14. Execute **./controls vcan0** to start the CANBus Control Panel. You will see the CANBus Control Panel interface as shown in the screenshot.

15.



16. 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/**.
17. Execute **cansniffer -c vcan0** to start sniffing on the vcan0 interface. Leave this sniffer on.

18.

```

root@ubuntu-Virtual-Machine: /home/ubuntu/ICSim
root@ubuntu-Virtual-Machine: /h... x root@ubuntu-Virtual-Machine: /h... x root@ubuntu-Virtual-Machine: /h... x
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|ns | ID | data ... < vcan0 # l=20 h=100 t=500 slots=1 >
99999 | 143 | 6B 6B 00 D1 kk..
  
```

19. 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 -i vcan0**.
  20. 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**.
  21. Use the following keys to perform various functions
- |                                   |   |
|-----------------------------------|---|
| 22. <b>ICSim Functions</b>        | 23. <b>Keys</b>                           |
| 24. Accelerate                    | 25. Up arrow                              |
| 26. Left/Right Turn               | 27. Left arrow/ Right arrow               |
| 28. Unlock Rear Left/Right doors  | 29. Right Shift + X / Right Shift + Y     |
| 30. Unlock Front Left/Right doors | 31. Right Shift +A / Right Shift + B      |
| 32. Lock all doors                | 33. Hold Right Shift key + Tap Left Shift |
| 34. Unlock all doors              | 35. Hold Left Shift key + Tap Right Shift |

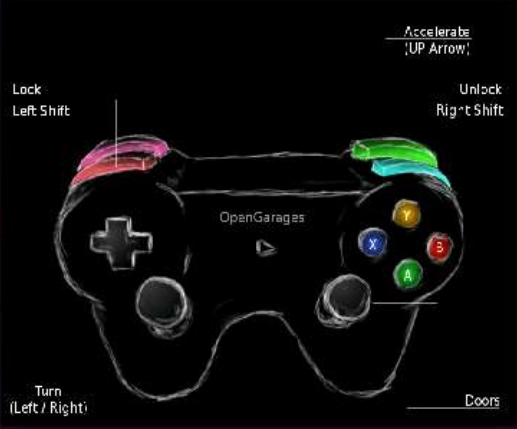
36.

Activities controls May 7 06:36

root@ubuntu-Virtual-Machine: /home/ubuntu/ICSim

```
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'
```

**CANBus Control Panel**



Accelerate 'UP Arrow'

Lock Left Shift


Unlock Right Shift

OpenGarages

Turn (Left / Right)

Doors

**IC Simulator**

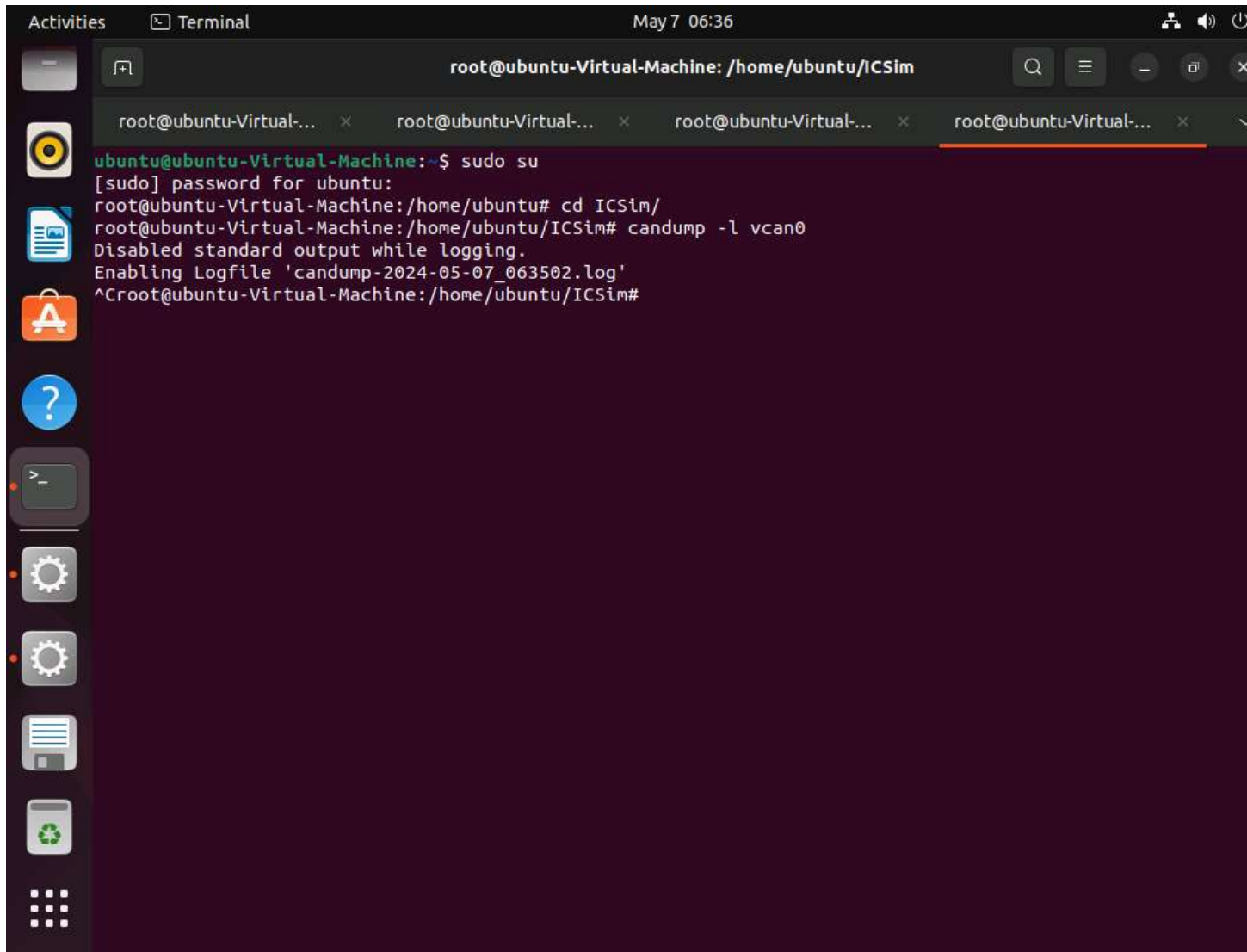


MPH

OpenGarages



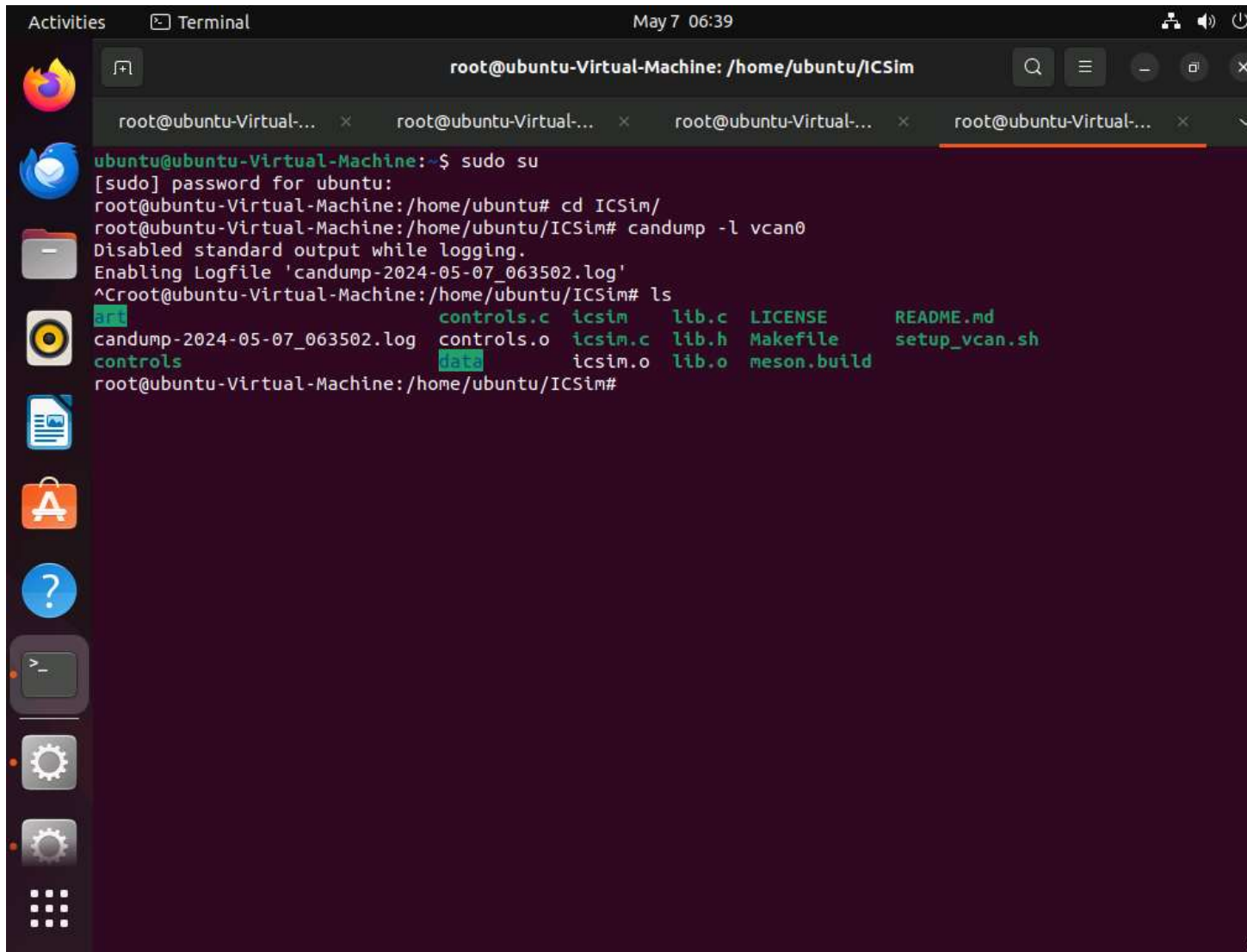
37.



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

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

39.

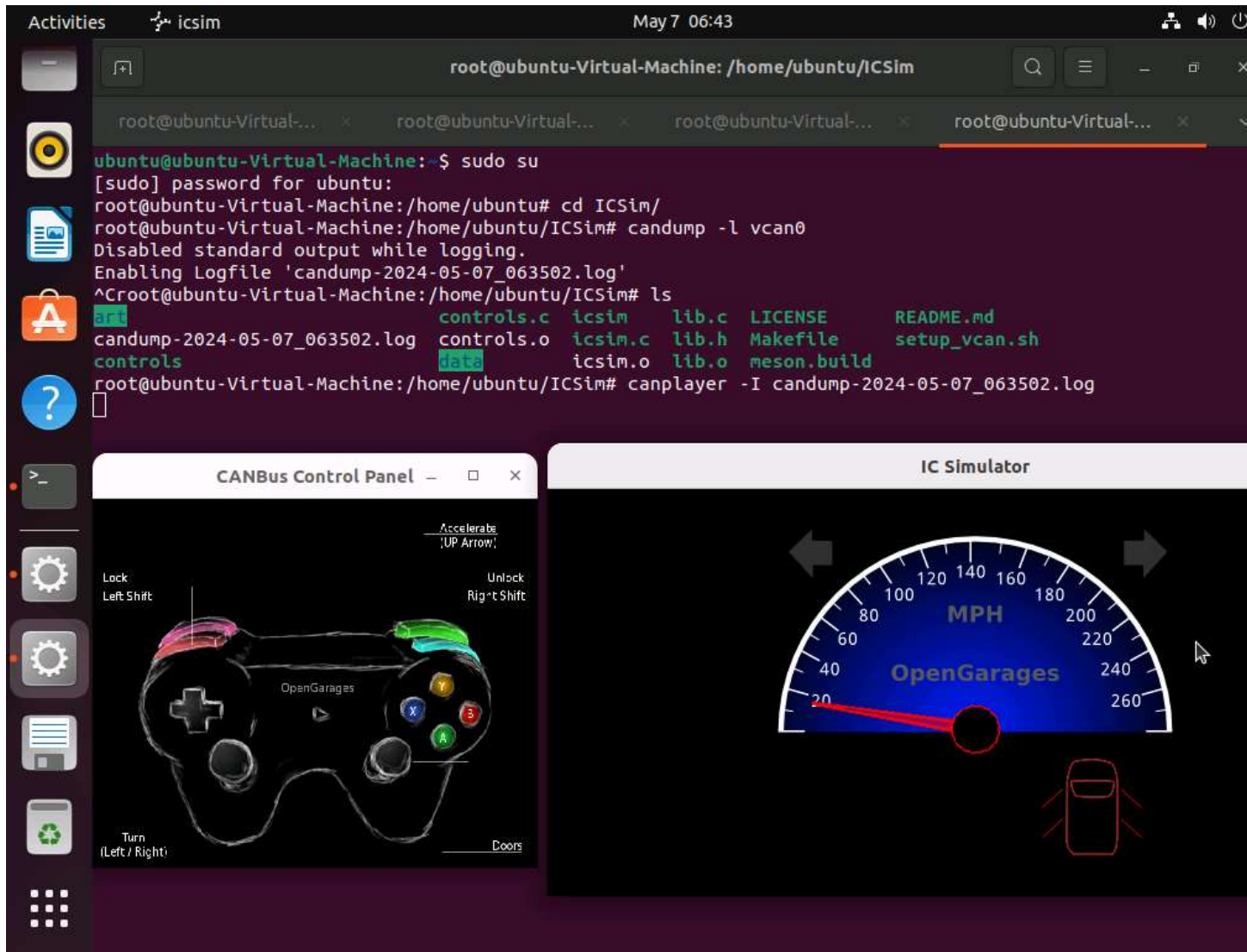


The screenshot shows a terminal window titled "Terminal" with the date "May 7 06:39". The terminal is running on a system named "root@ubuntu-Virtual-Machine" in the directory "/home/ubuntu/ICSim". The user "ubuntu" has executed the command "sudo su" and entered the password for "ubuntu". The prompt has changed to "root@ubuntu-Virtual-Machine:/home/ubuntu#". The user has then executed "cd ICSim/" and "candump -l vcan0". The output shows that standard output is disabled while logging and a logfile "candump-2024-05-07\_063502.log" is being enabled. The user has then executed "ls" and the output shows a list of files and directories: "candump-2024-05-07\_063502.log", "controls", "controls.c", "controls.o", "data", "icsim", "icsim.c", "icsim.o", "lib.c", "lib.h", "lib.o", "LICENSE", "Makefile", "meson.build", "README.md", and "setup\_vcan.sh".

```
root@ubuntu-Virtual-Machine: /home/ubuntu/ICSim
root@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
candump-2024-05-07_063502.log  controls.c  icsim      lib.c      LICENSE    README.md
controls                       controls.o  icsim.c    lib.h      Makefile   setup_vcan.sh
data                          icsim.o    lib.o      meson.build
```

40. Now, to perform replay attack, run **canplayer -l candump-2024-05-07\_063502.log** and press enter.
41. 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.
42. The log file name might vary while performing lab.

43.



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

45. 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.

Score

- Check this box to confirm completion of this module.

Previous<sup>9</sup>Next<sup>10</sup>

11

40 Minutes Remaining

Thumbnail screenshot of virtual machineLab52683430-Windows 11

Previous: Lab 2: Capture and Analyze IoT Device Traffic

Next

0/101 (0%) Tasks Complete

Windows 11

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Admin<sup>12</sup>

Password

Pa\$\$w0rd<sup>13</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683430-Windows Server 2022

Windows Server 2022

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Administrator<sup>14</sup>

Password

Pa\$\$w0rd<sup>15</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683430-Windows Server 2019

Windows Server 2019

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Administrator<sup>16</sup>

Password

Pa\$\$w0rd<sup>17</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683430-Parrot Security

Parrot Security

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Attacker<sup>18</sup>

Password

toor<sup>19</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683430-Ubuntu

Ubuntu

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Ubuntu<sup>20</sup>

Password

toor<sup>21</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

**Help**

---

Type Text

Type Text

Type Text

Type Text

Type Text

Type Text

Type Text

Type Text

Type Text

Type Text

## Support Information

ID	52683430
Host	EU-HV20
Datacenter	EU North (London)

## FAQs

[Frequently asked questions about the lab interface](#)

## Other Help Options

[Submit a Support Request](#)

Powered by [Skillable](#)•[Review Us](#)

## Notifications

## Settings

### Text Size

100 Standard  
150 Large Text  
200 Extra Large Text

---

### Color Mode

- Light
- Dark
- High Contrast

---

### Actions

[Split Windows](#)

Close Window

Close Window