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

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

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

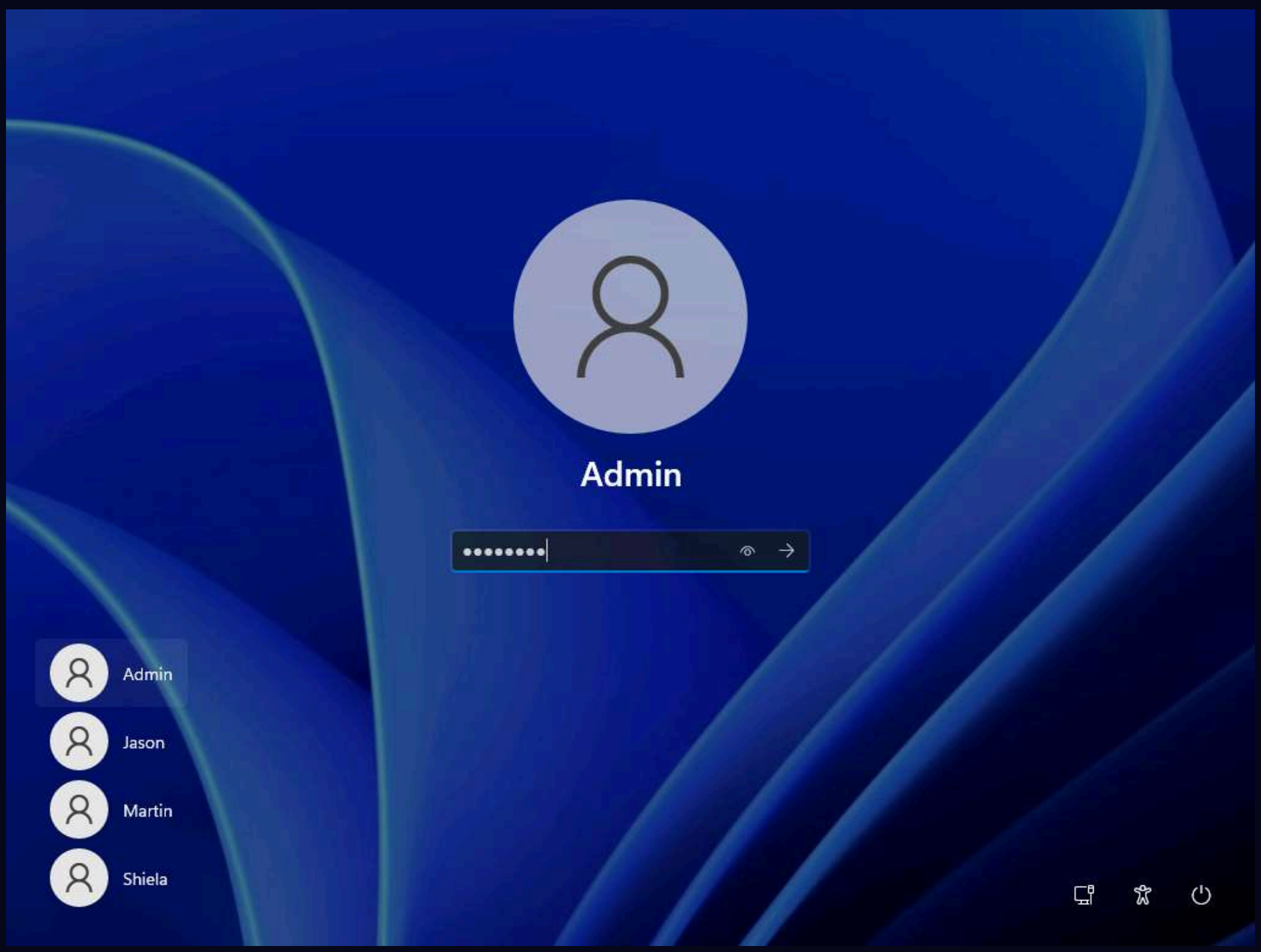
The information regarding the target IoT and OT devices can be acquired using various online sources such as Whois domain lookup, advanced Google hacking, and Shodan search engine. The gathered information can be used to scan the devices for vulnerabilities and further exploit them to launch attacks.

> Note: 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. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine, click **Ctrl+Alt+Del**.
2. By default, **Admin** user profile is selected, type **Pa$$w0rd** in the **Password** field and press **Enter** to login.

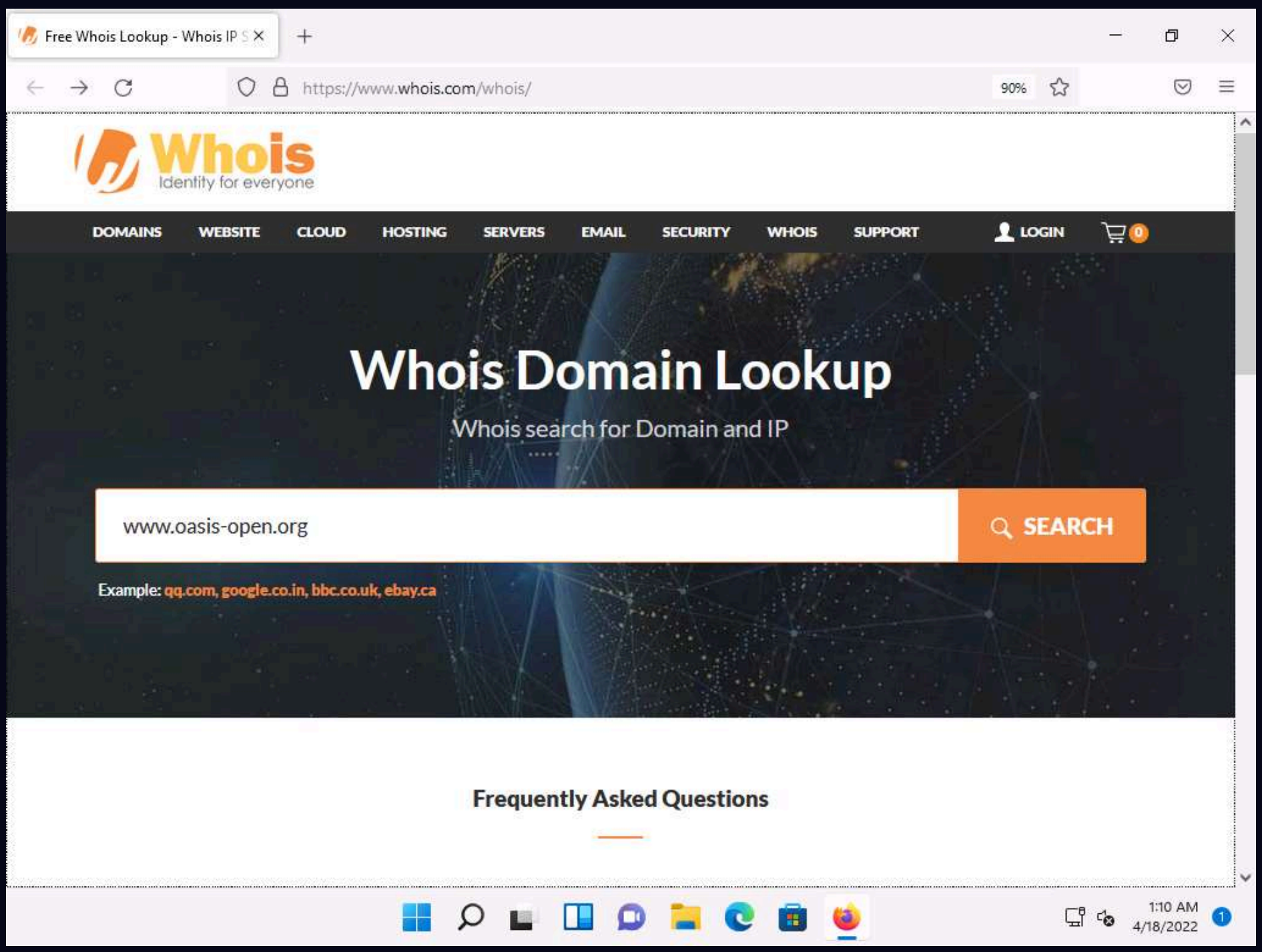
> Note: If **Welcome to Windows** wizard appears, click **Continue** and in **Sign in with Microsoft** wizard, click **Cancel**.
>
> Note: Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.



3. Launch any browser, here, we are using **Mozilla Firefox**. In the address bar of the browser place your mouse cursor and type **https://www.whois.com/whois/** and press **Enter**.

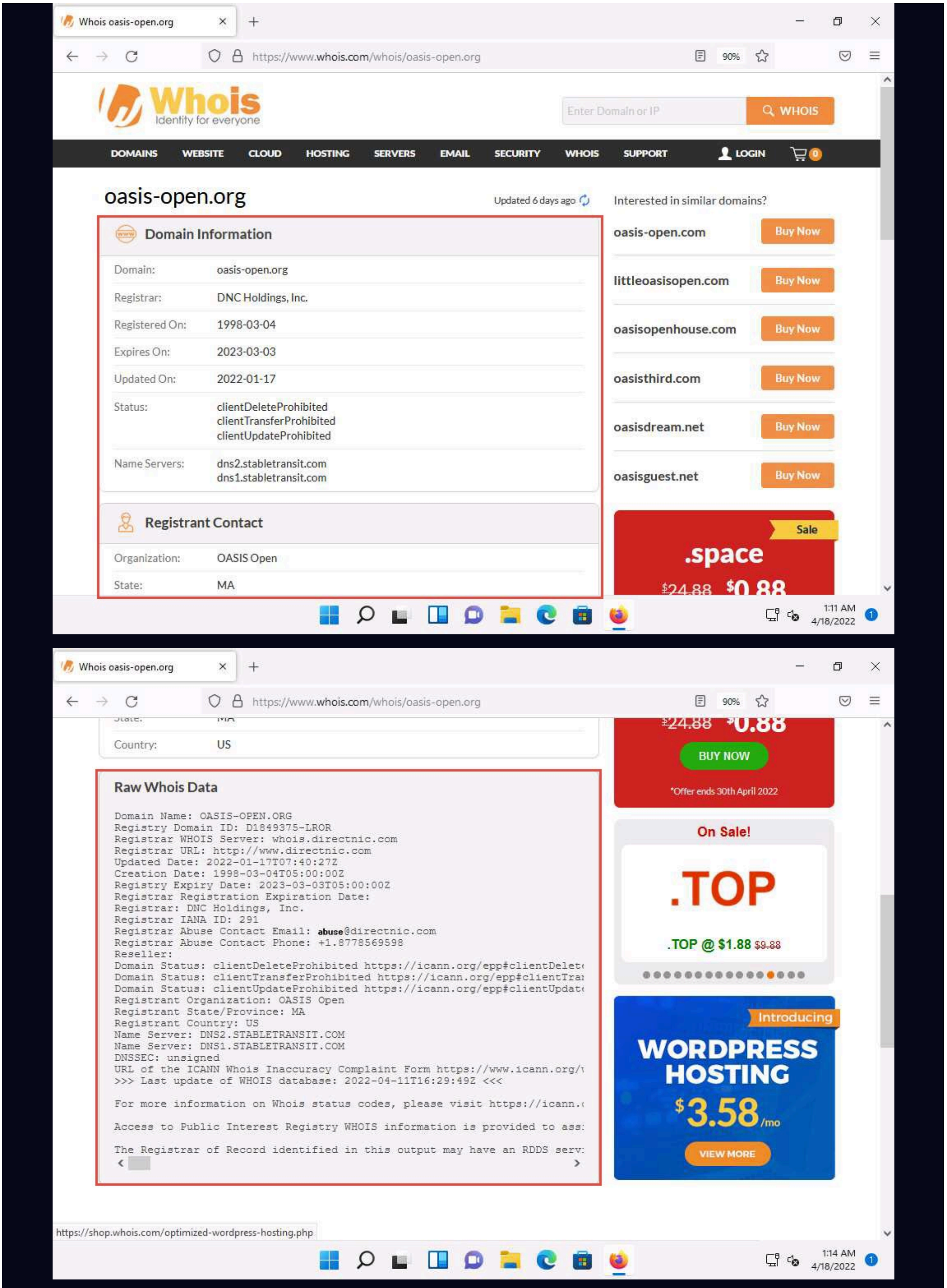
4. The **Whois Domain Lookup** page appears; type **www.oasis-open.org** in the search field and click **SEARCH**.

Note: 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.



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

Note: 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.

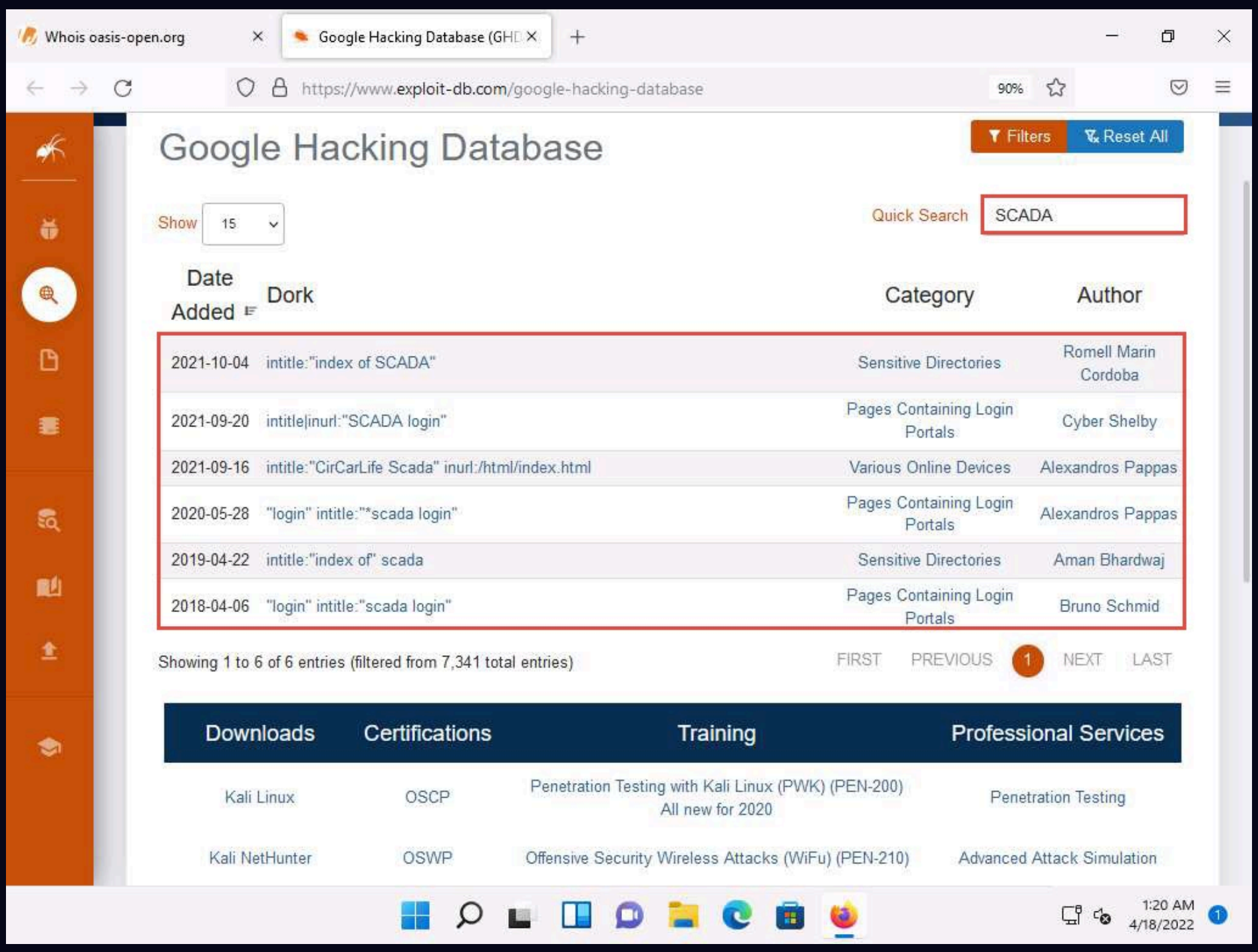


Note: Whois lookup reveals available information on a hostname, IP address, or domain.

6. Now, open a new tab, and type **https://www.exploit-db.com/google-hacking-database** in the address bar, and press **Enter**.

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

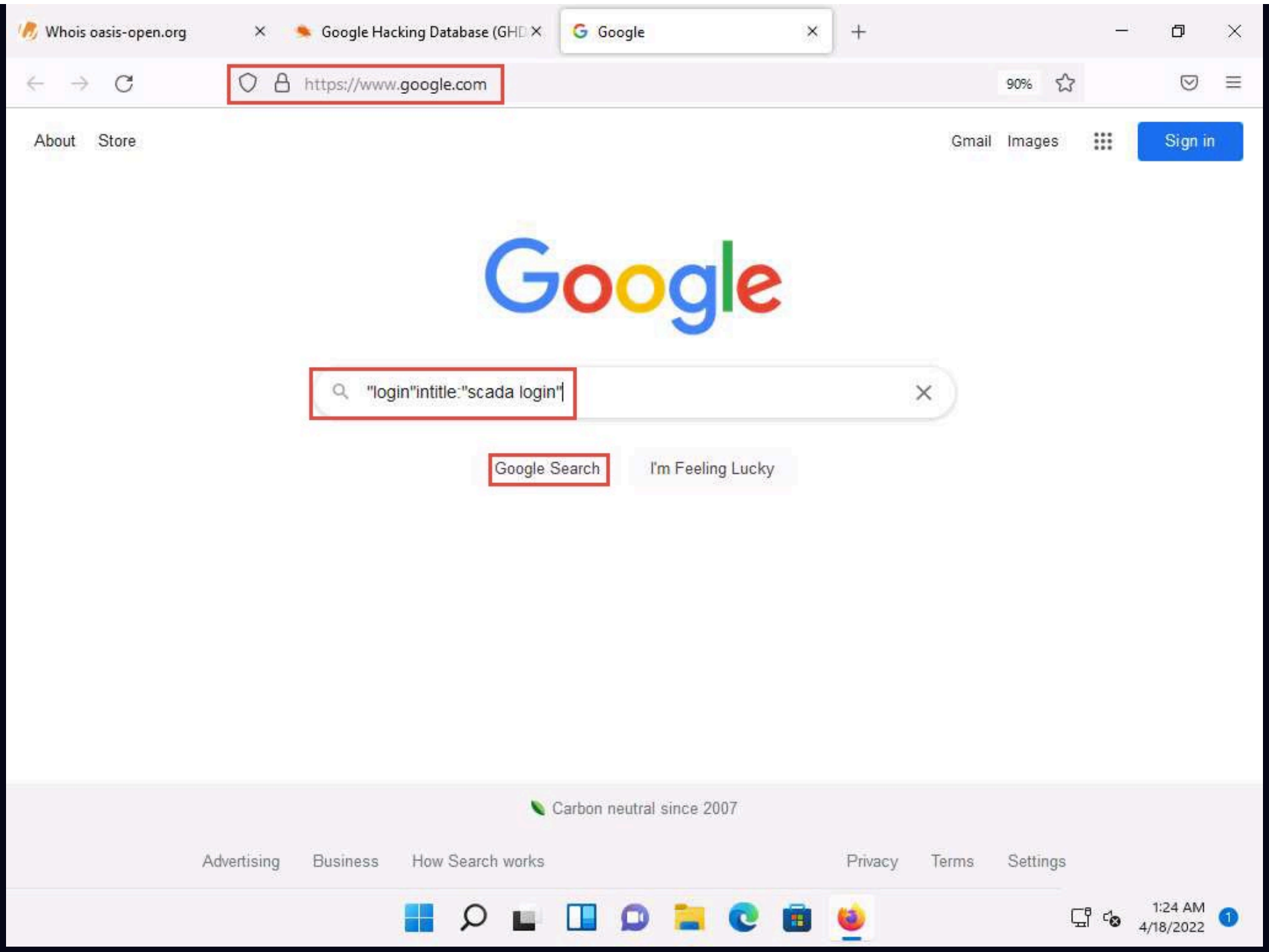
8. The result appears, which displays the Google dork related to SCADA, as shown in the screenshot.



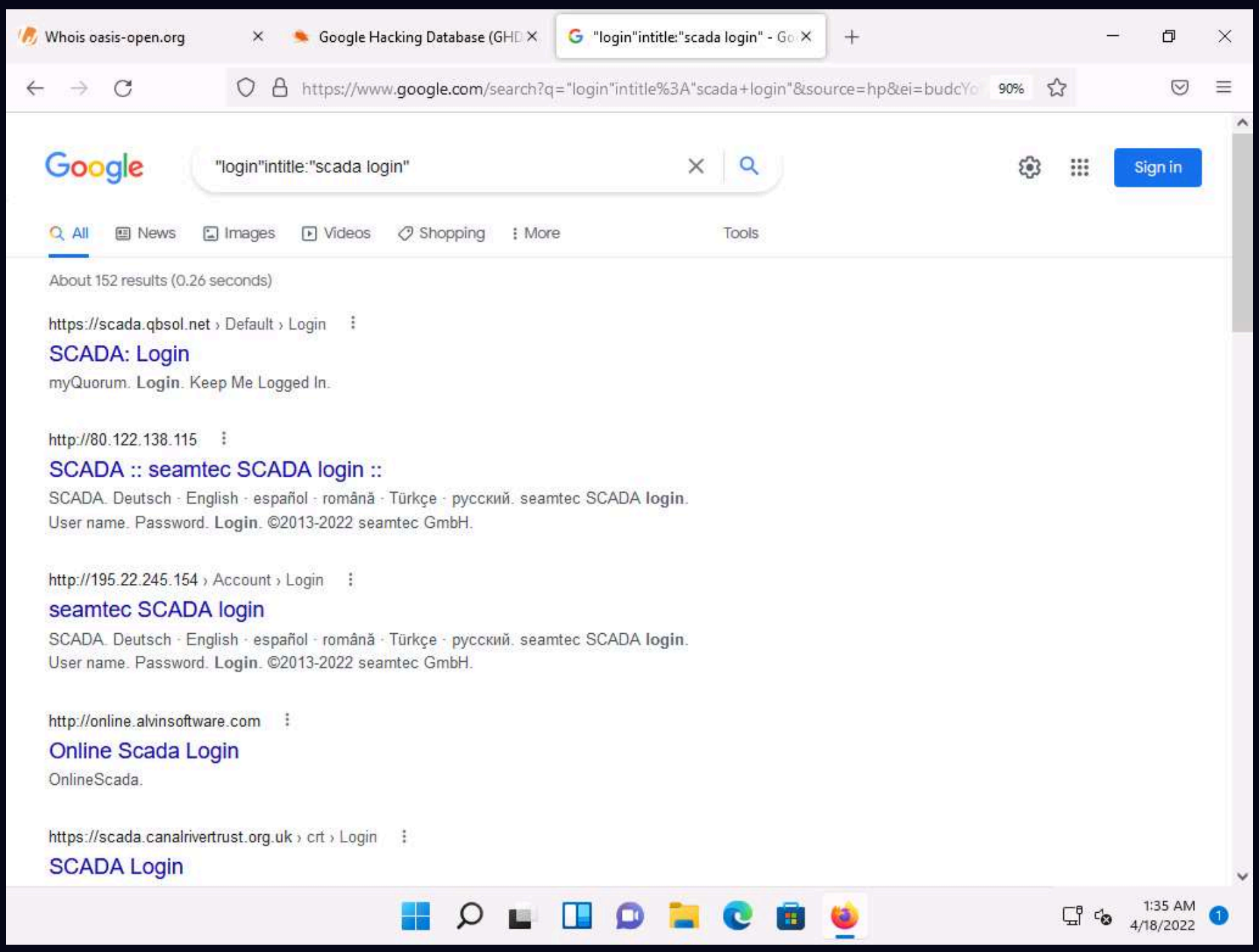
9. Now, we will use the dorks obtained in the previous step to query results in Google.

10. Open a new tab and type **https://www.google.com** in the address bar, and press **Enter**.

11. In the search field, type **"login" intitle:"scada login**" and click the **Google Search** button.



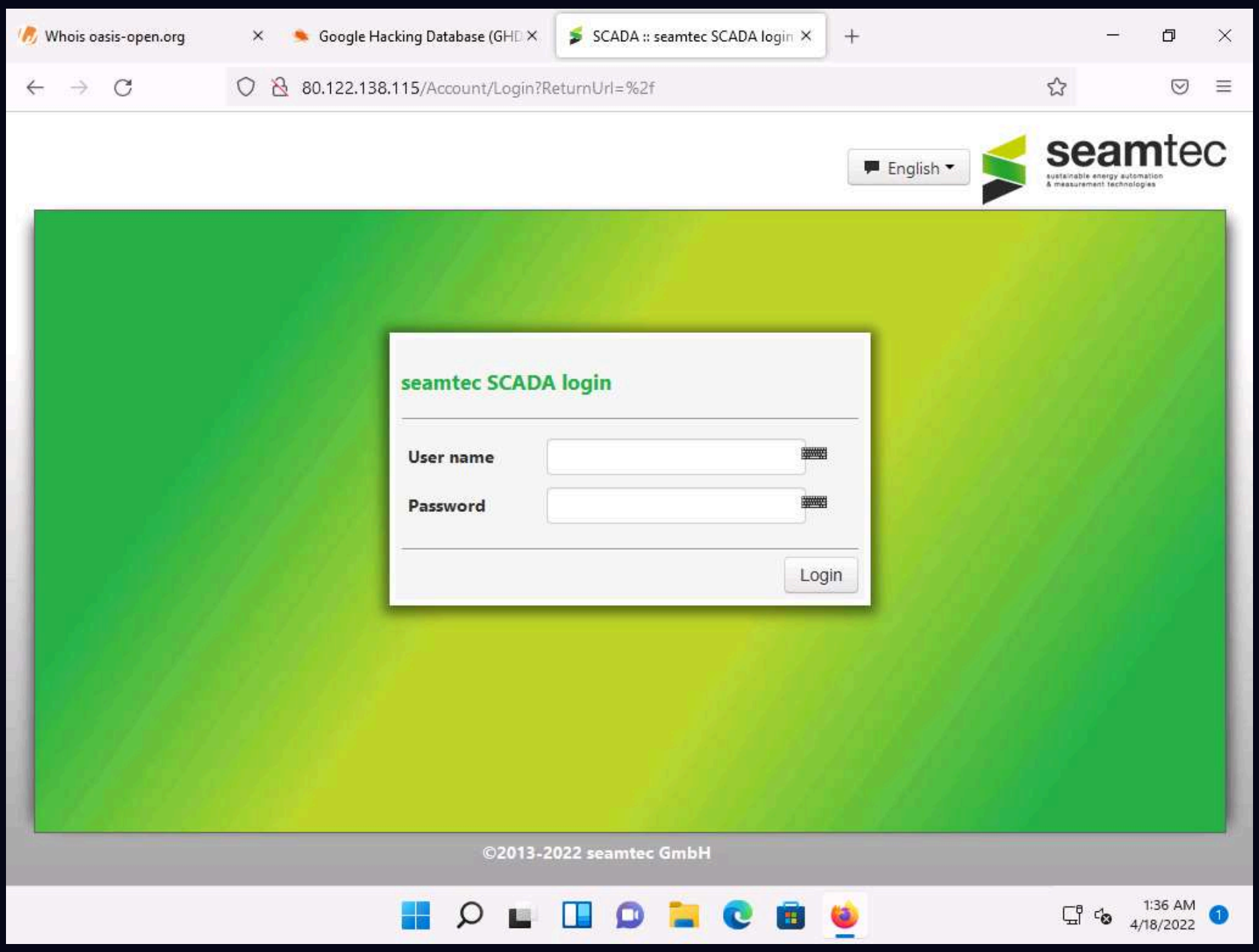
12. The search result appears; click any link (here, **SCADA :: seamtec SCADA login ::**).



Note: 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.

13. The **seamtec SCADA login** page appears, as shown in the screenshot.

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

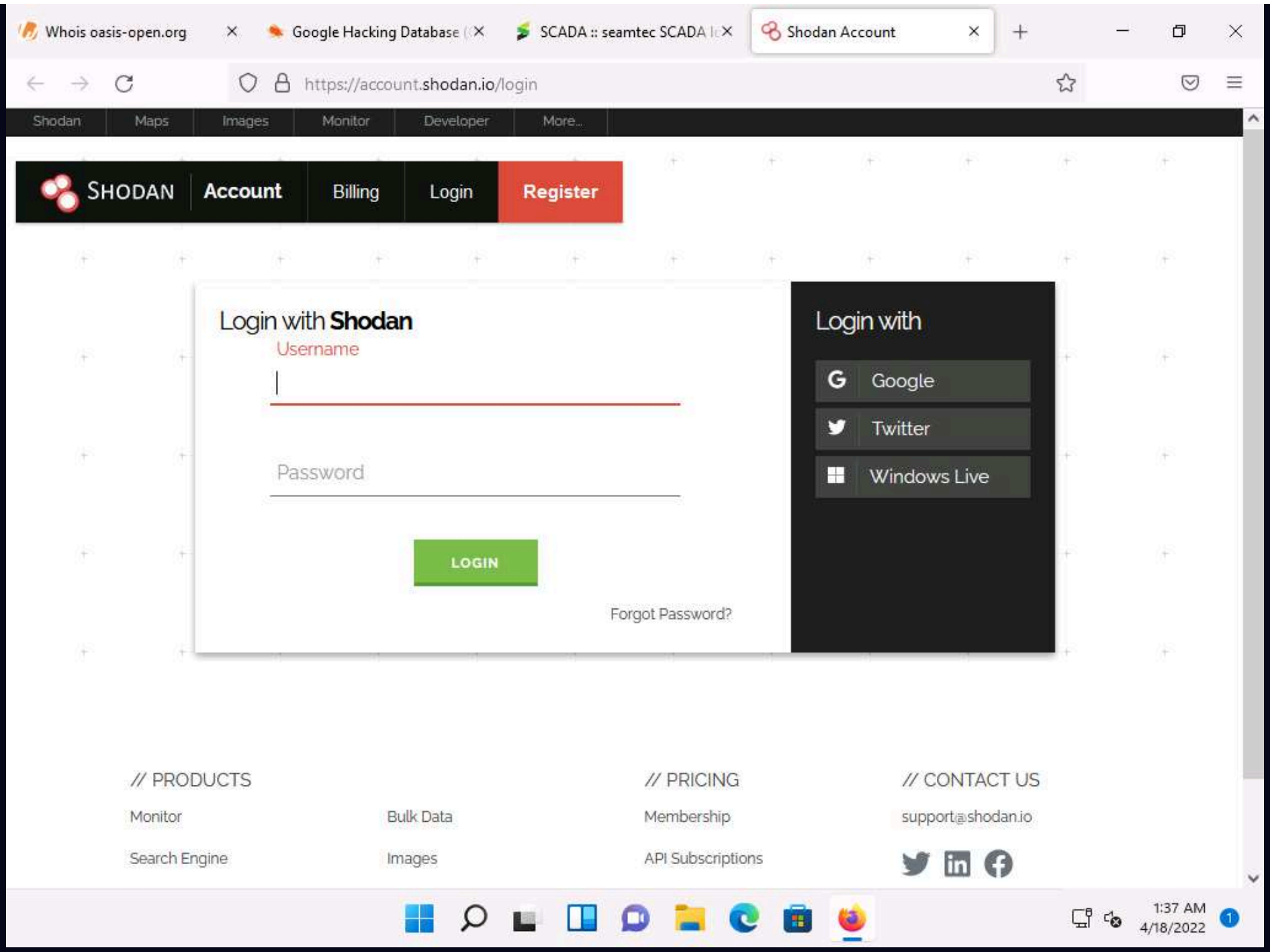


14. Similarly, you can use advanced search operators such as **intitle:"index of" scada** to search sensitive SCADA directories that are exposed on sites.

15. Now, in the browser window, open a new tab type **https://account.shodan.io/login** in the address bar, and press **Enter**.

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

Note: Go to the **Register** option to register yourself if you do not have an existing account.

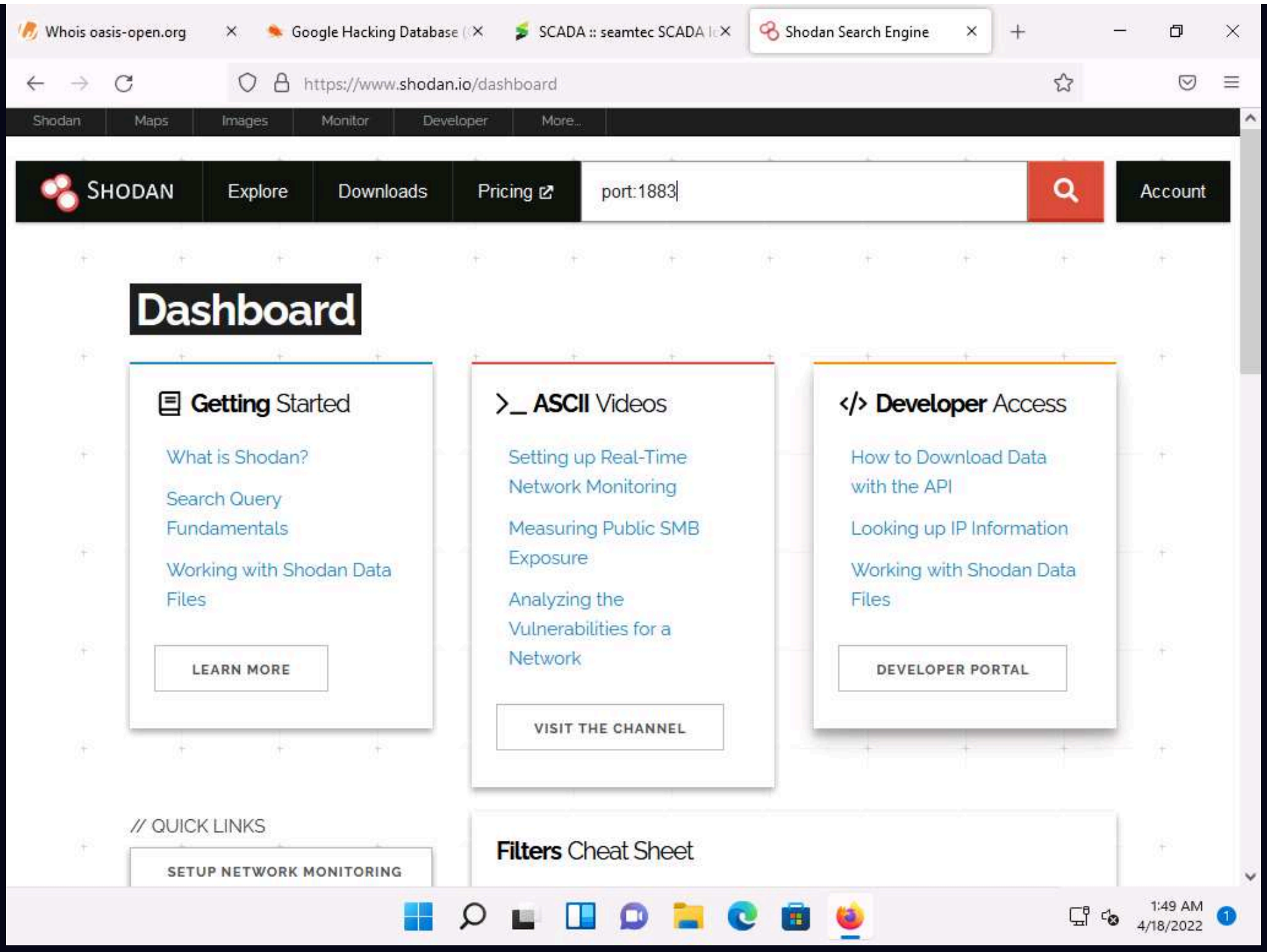


17. The **Account Overview** page appears, which displays the account-related information.

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

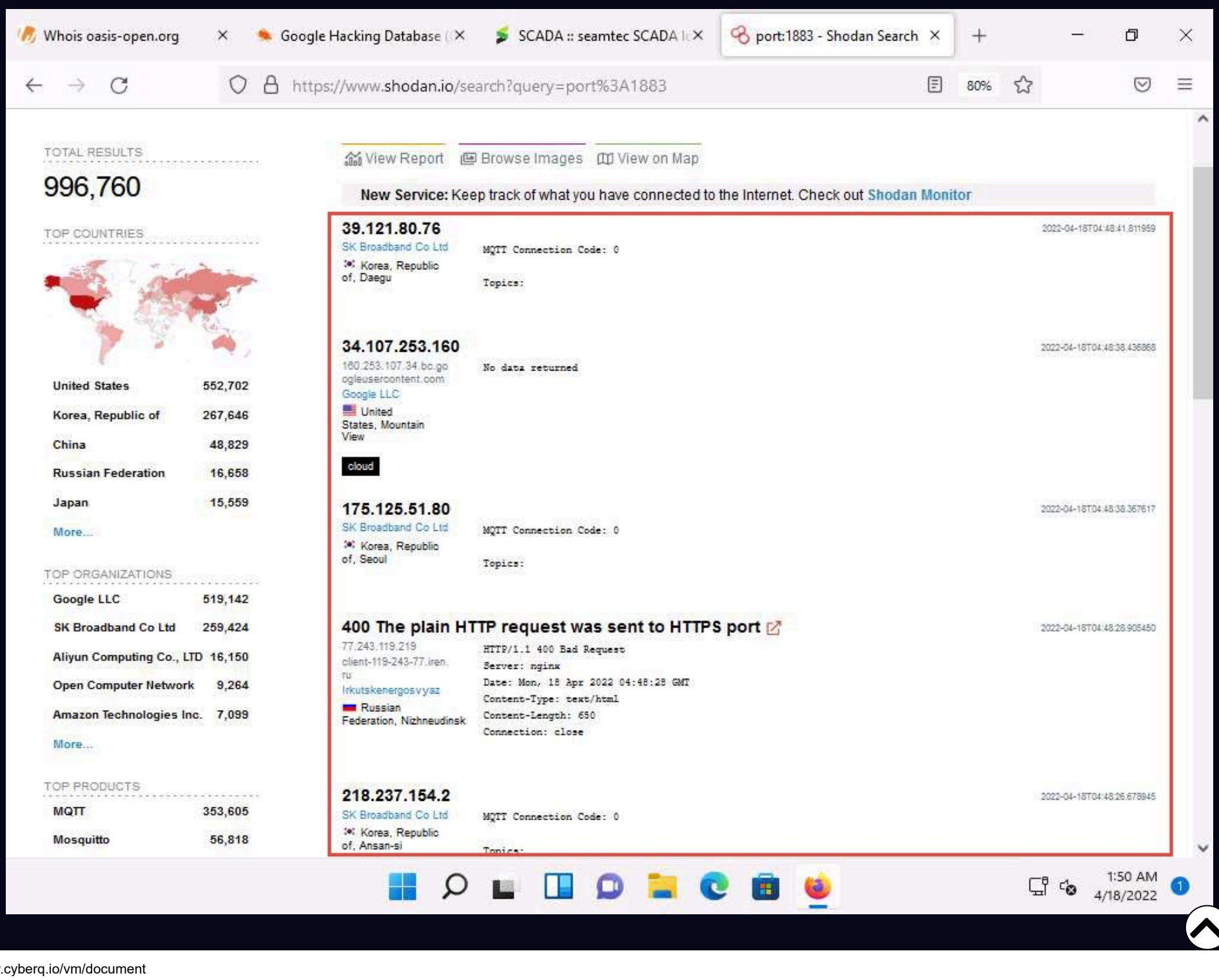
18. The **Shodan** main page appears; type **port:1883** in the address bar and press **Enter**.

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

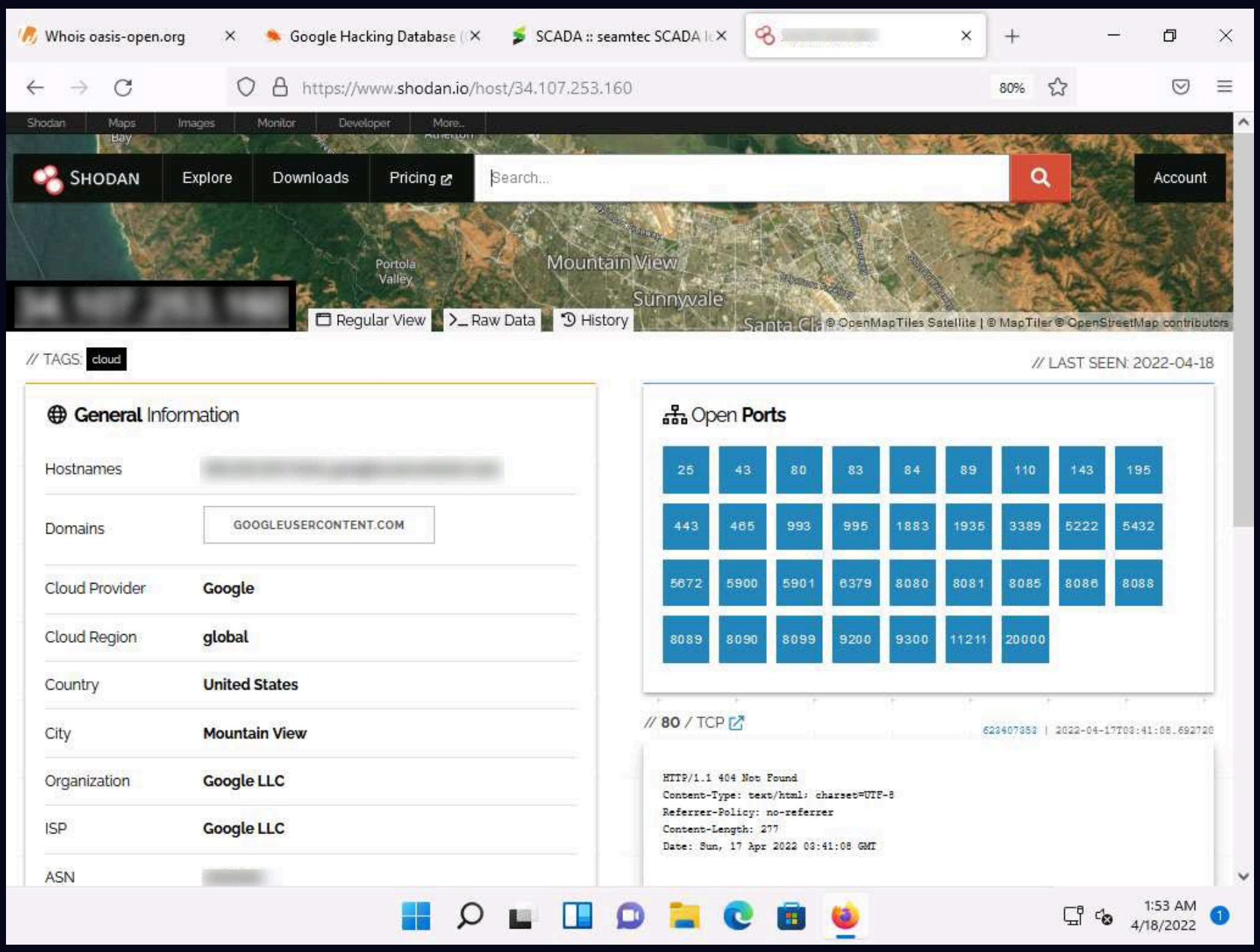


19. The result appears, displaying the list of IP addresses having port 1883 enabled, as shown in the screenshot.

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



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



22. 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"
23. 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.
24. 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.
25. Close all open windows and document all the acquired information.

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.

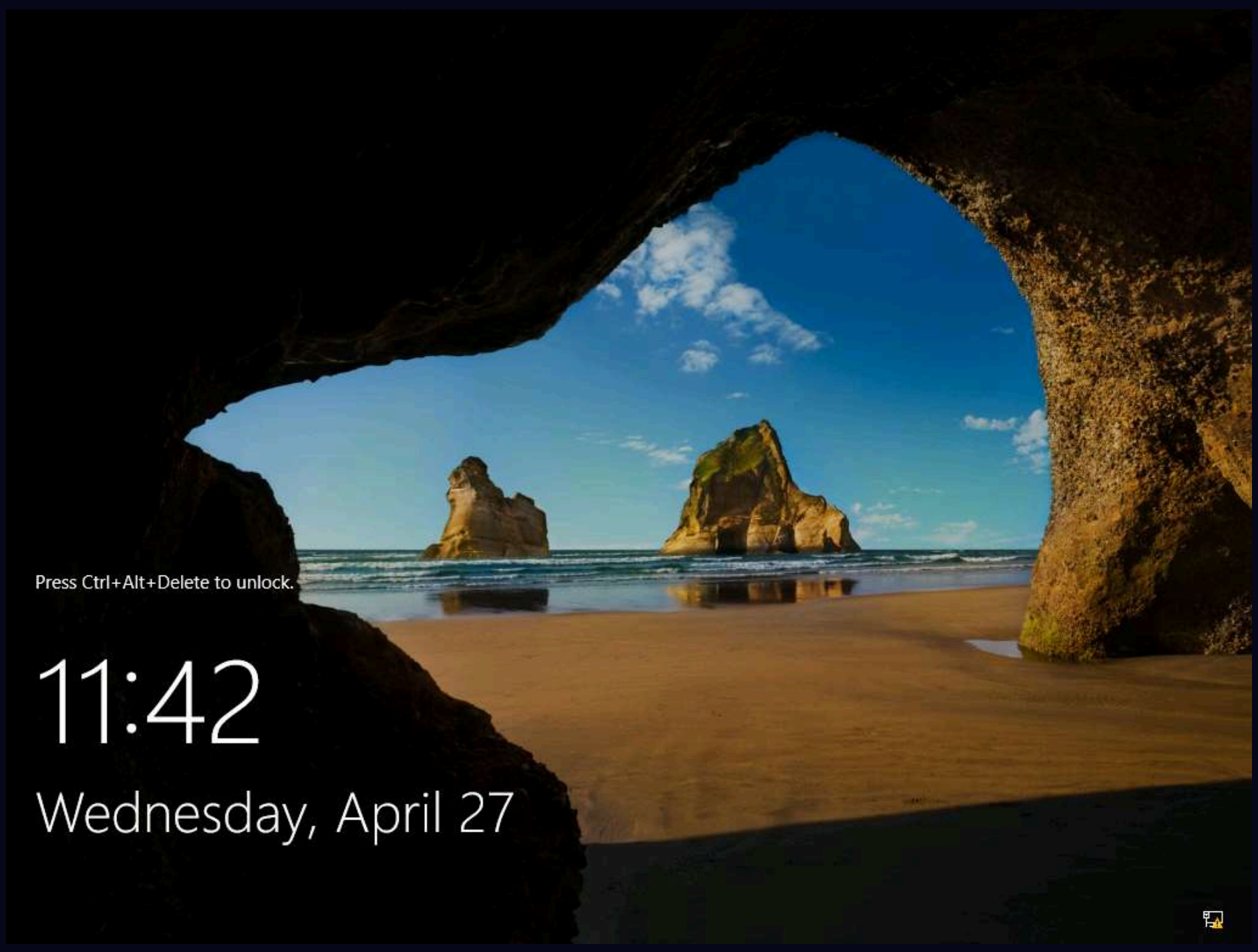
Task 1: Capture and Analyze IoT Traffic using Wireshark

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.

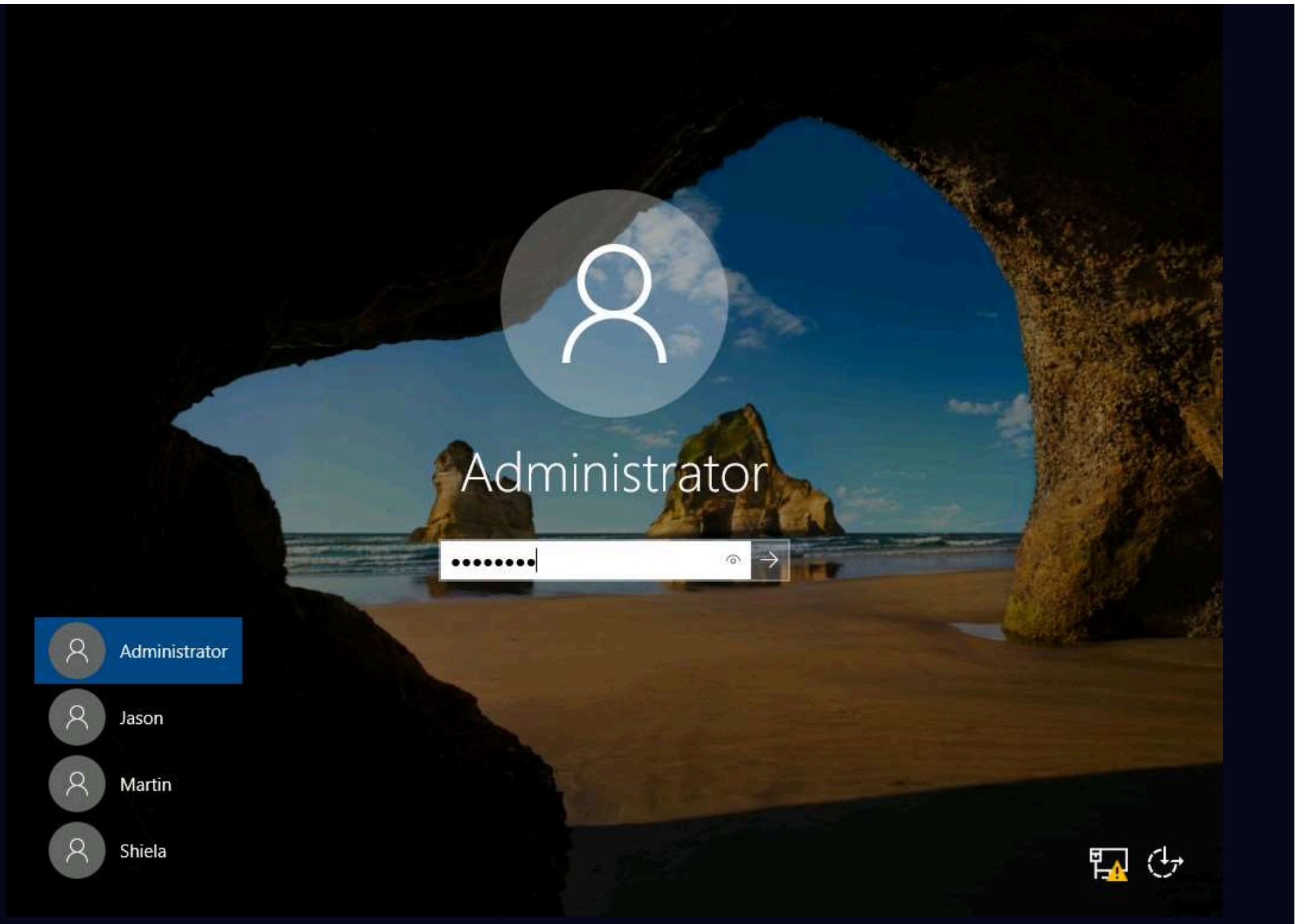
MQTT is a lightweight messaging protocol that uses a publish/subscribe communication pattern. Since the protocol is meant for devices with a low-bandwidth, it is considered ideal for machine-to-machine (M2M) communication or IoT applications. We can create virtual IoT devices over the virtual network 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 **CEHv12 Windows Server 2019** to launch **Windows Server 2019** machine, and then click **Ctrl+Alt+Del** link to login.

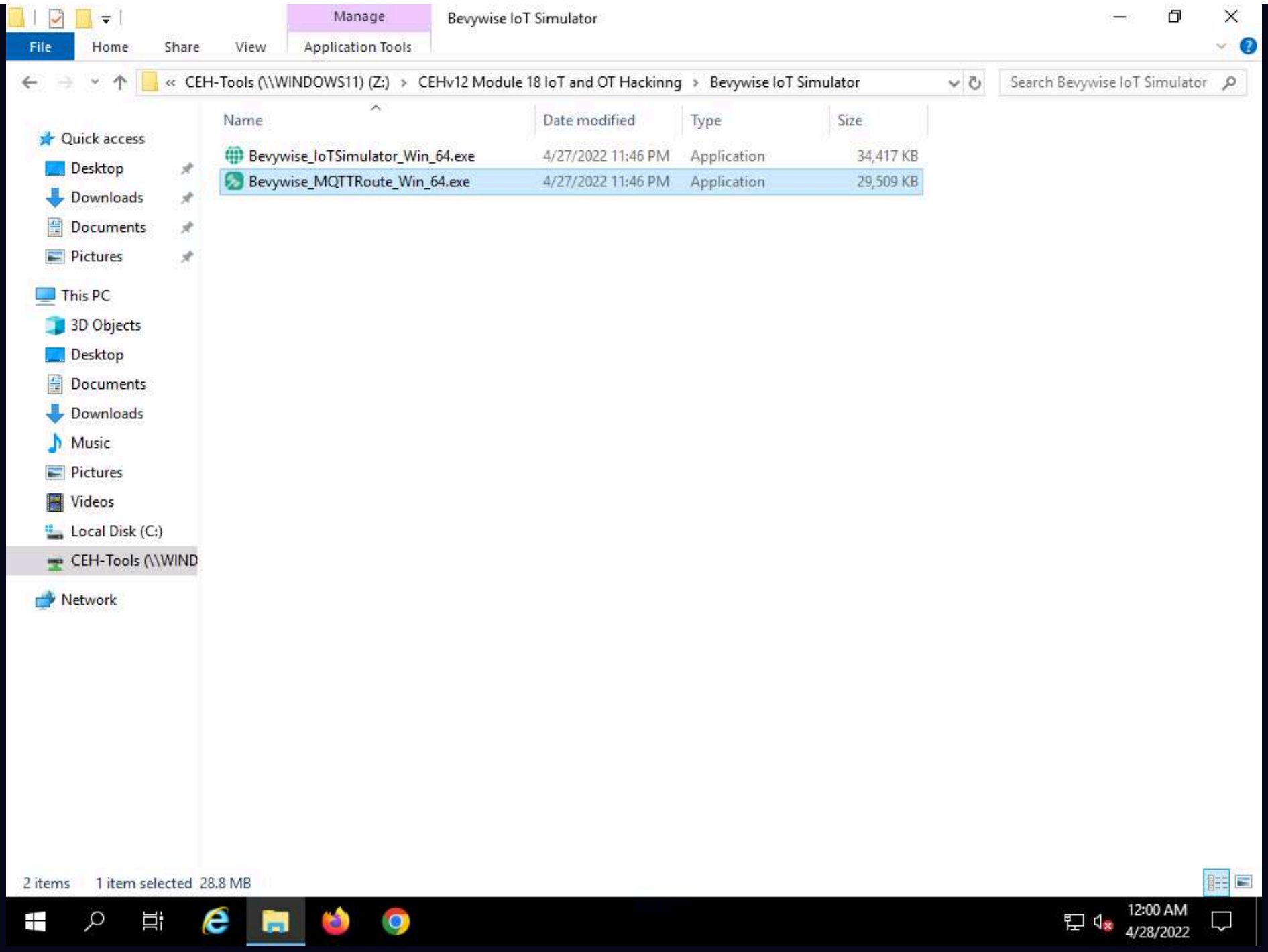


2. By default **Administrator ** account is selected, type **Pa$$w0rd** in the Password field and press **Enter** to login.



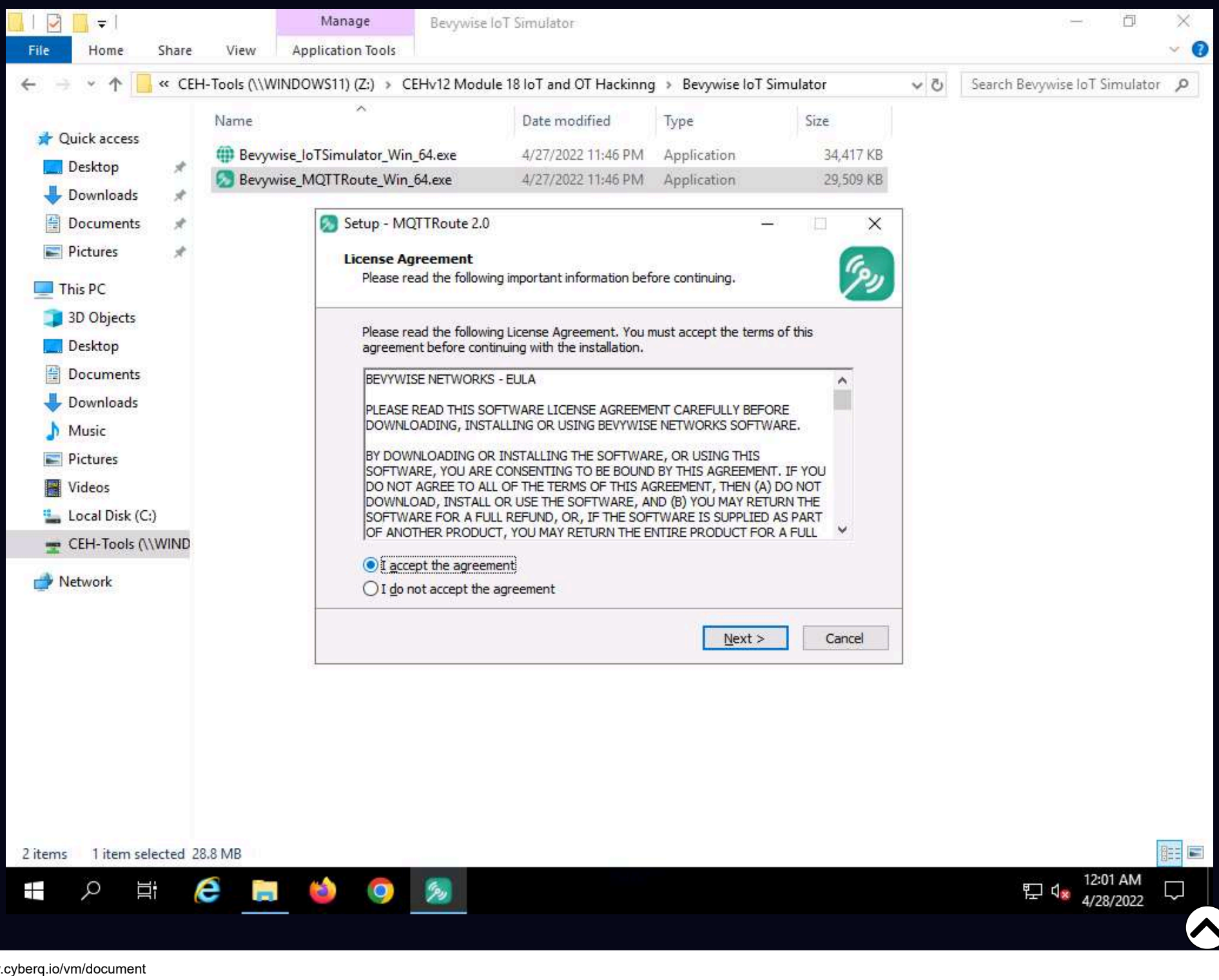
Note: If the network screen appears, click **Yes**.

3. Navigate to **Z:\CEH-Tools\CEHv12 Module 18 IoT and OT Hacking\Bevywise IoT Simulator** folder and double-click on the **Bevywise_MQTTRoute_Win_64.exe** file.



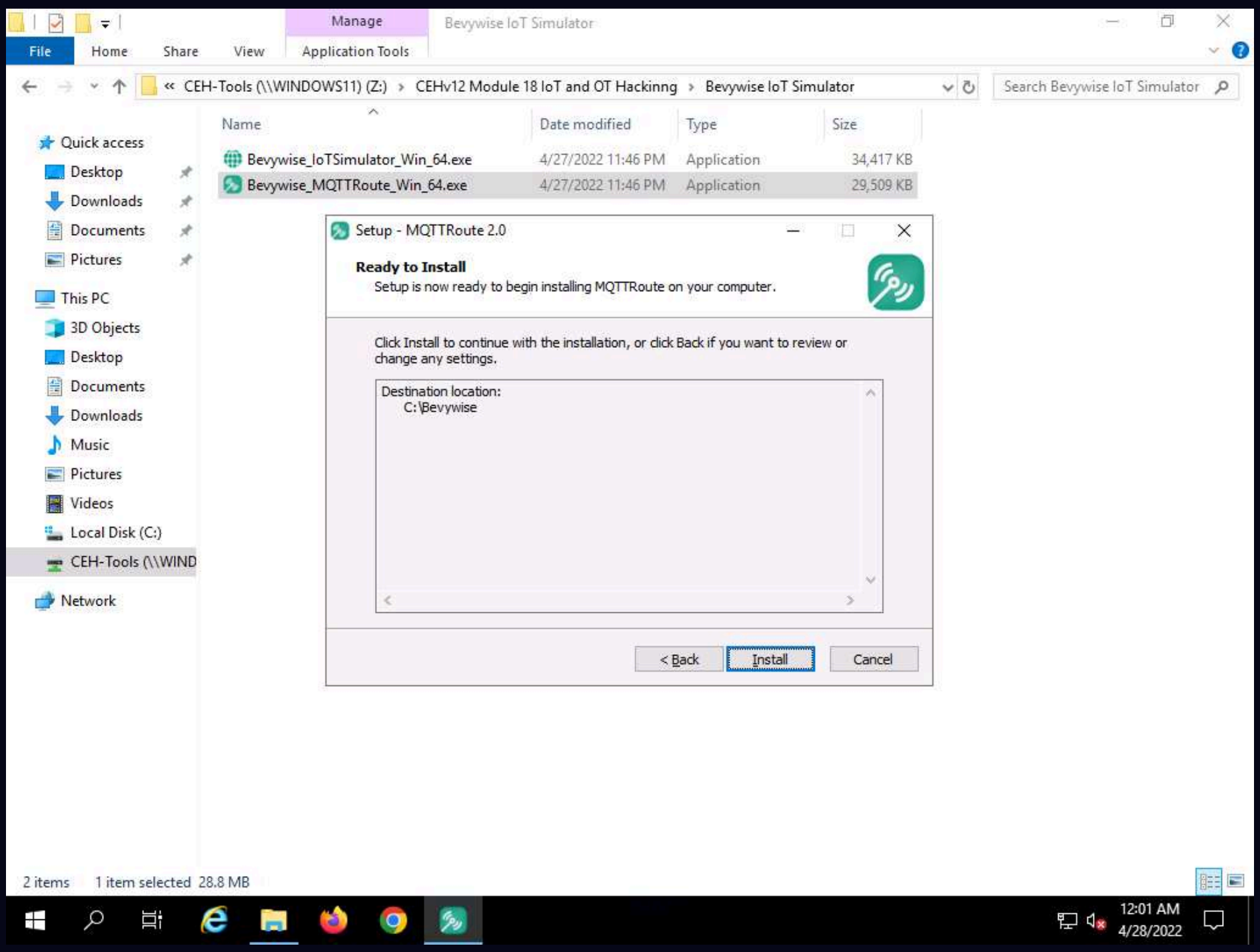
4. The **Open File - Security Warning** popup appears. Click **Run**.

5. The **Setup – MQTTRoute 2.0** window opens. Select **I accept the agreement** and click on **Next**.

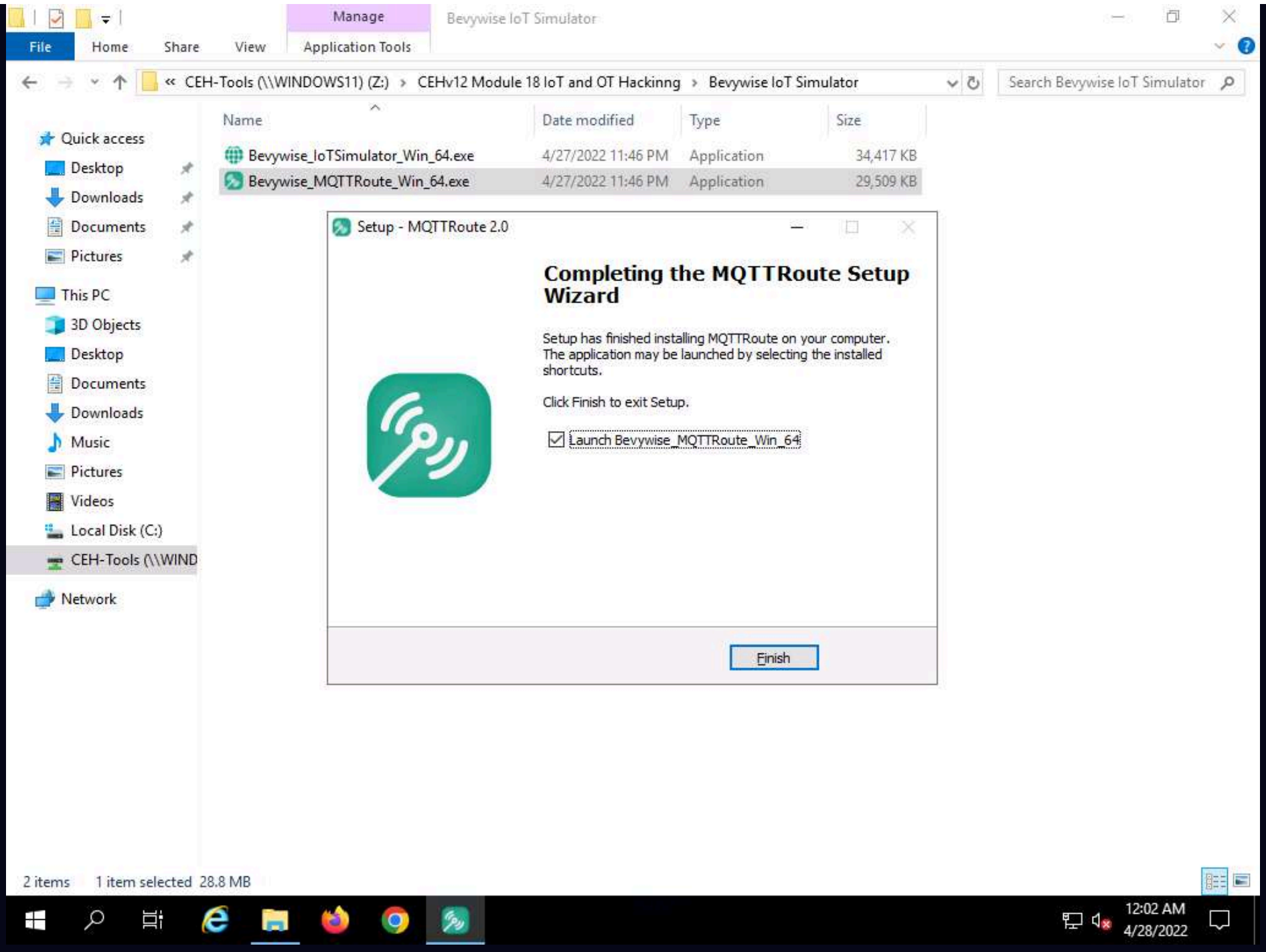


6. **Select Destination Location** page appears, without making any changes to the default installation location, click on **Next**.

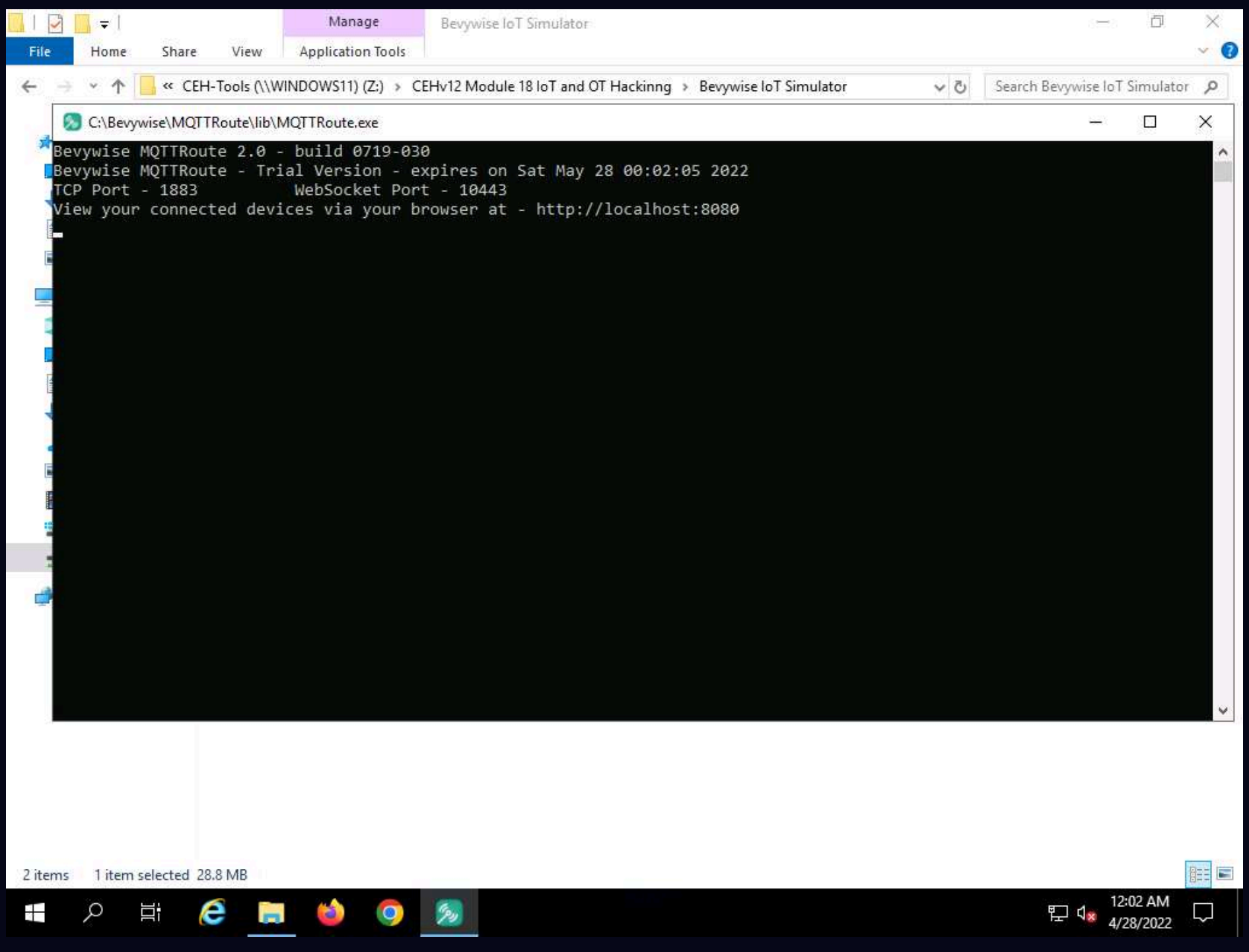
7. In the next window, click **Install** to complete the installation.



8. The installation completes, click on **Finish**. Ensure that **Launch Bevywise_MQTTRoute_Win_64** is checked.



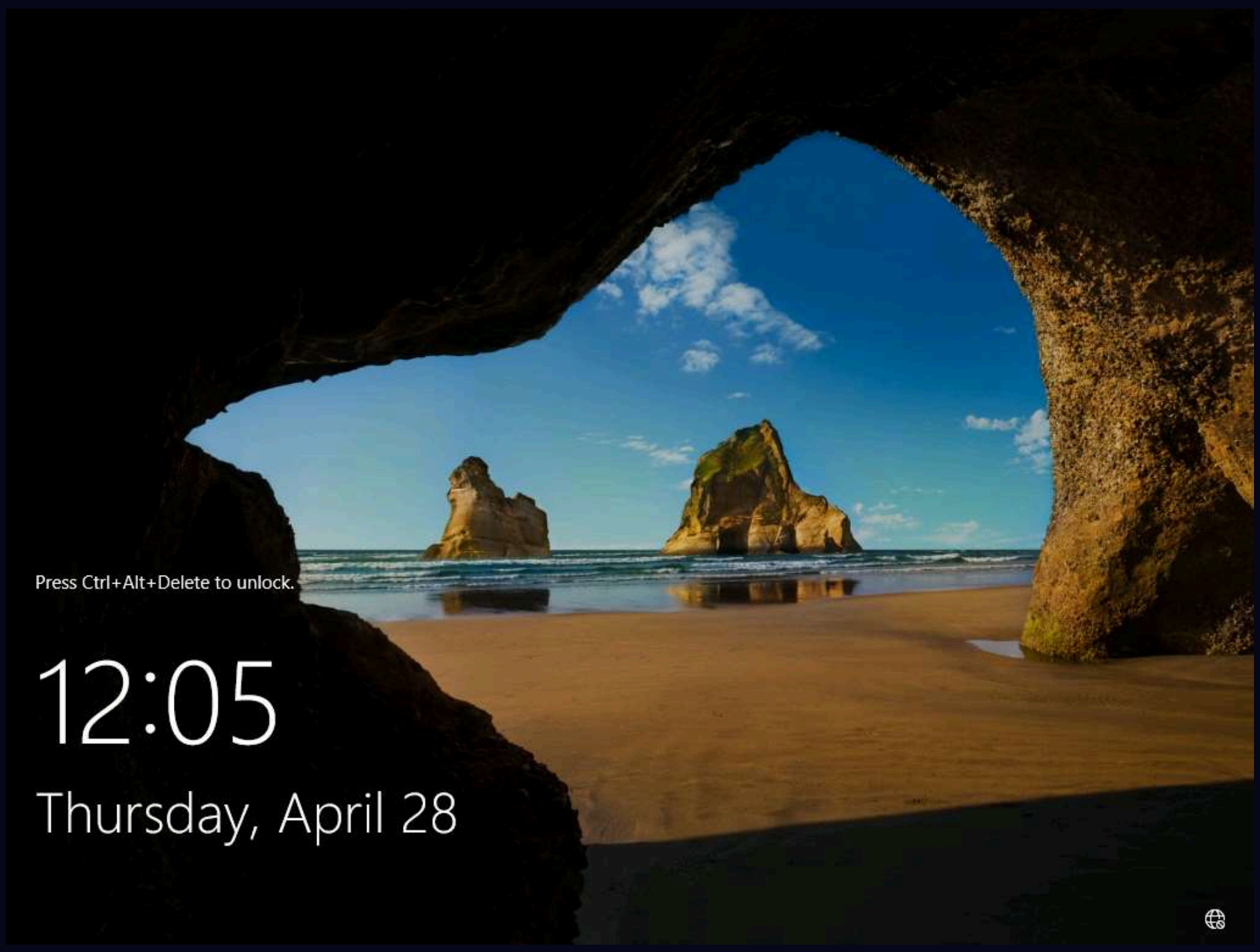
9. The MQTTRoute will execute and the command prompt will appear. You can see the **TCP** port using **1883**.



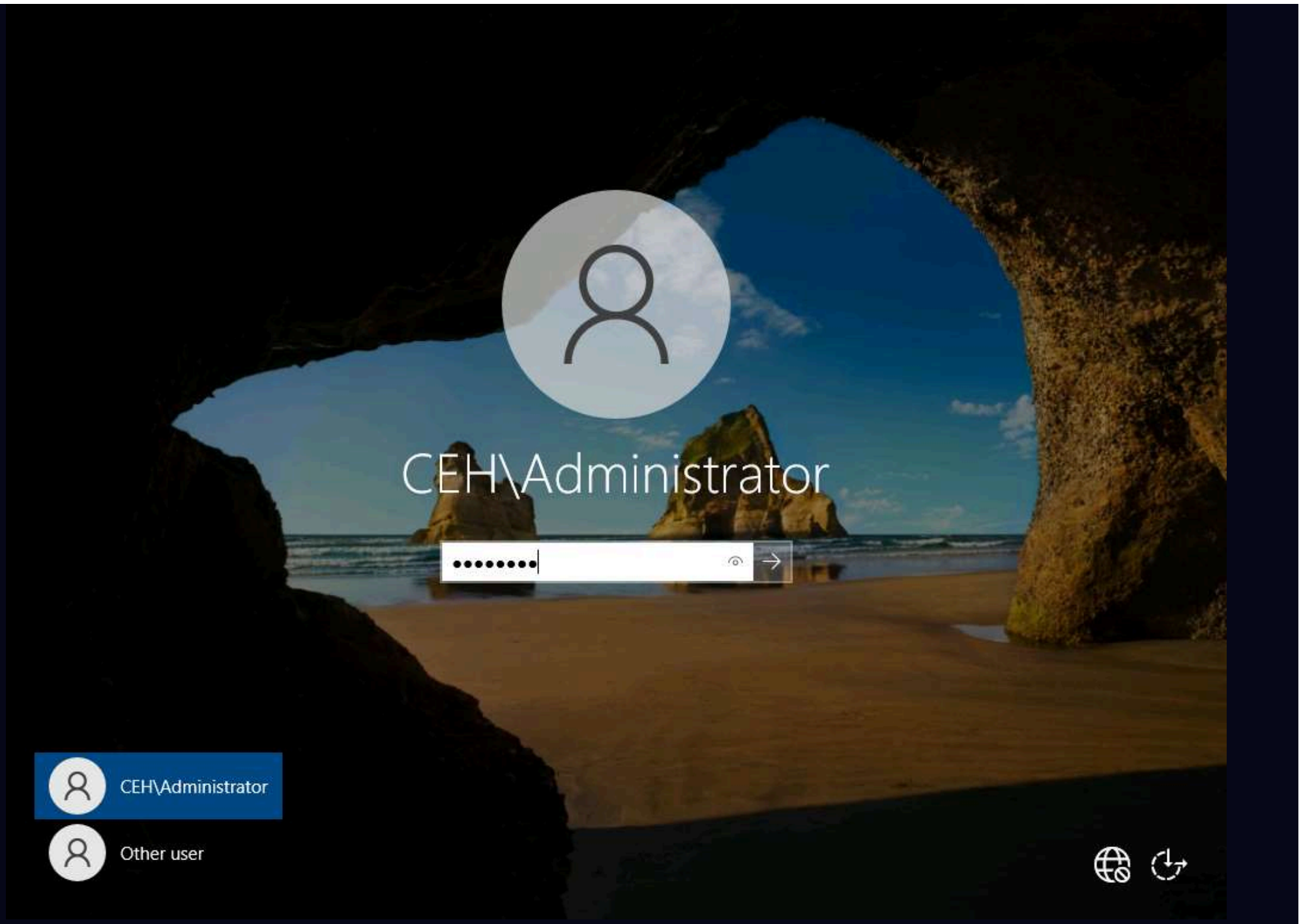
10. We have installed MQTT Broker successfully and leave the Bevywise MQTT **running**.

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

12. Click **CEHv12 Windows Server 2022** to launch **Windows Server 2022** machine. Click **Ctrl+Alt+Del**.

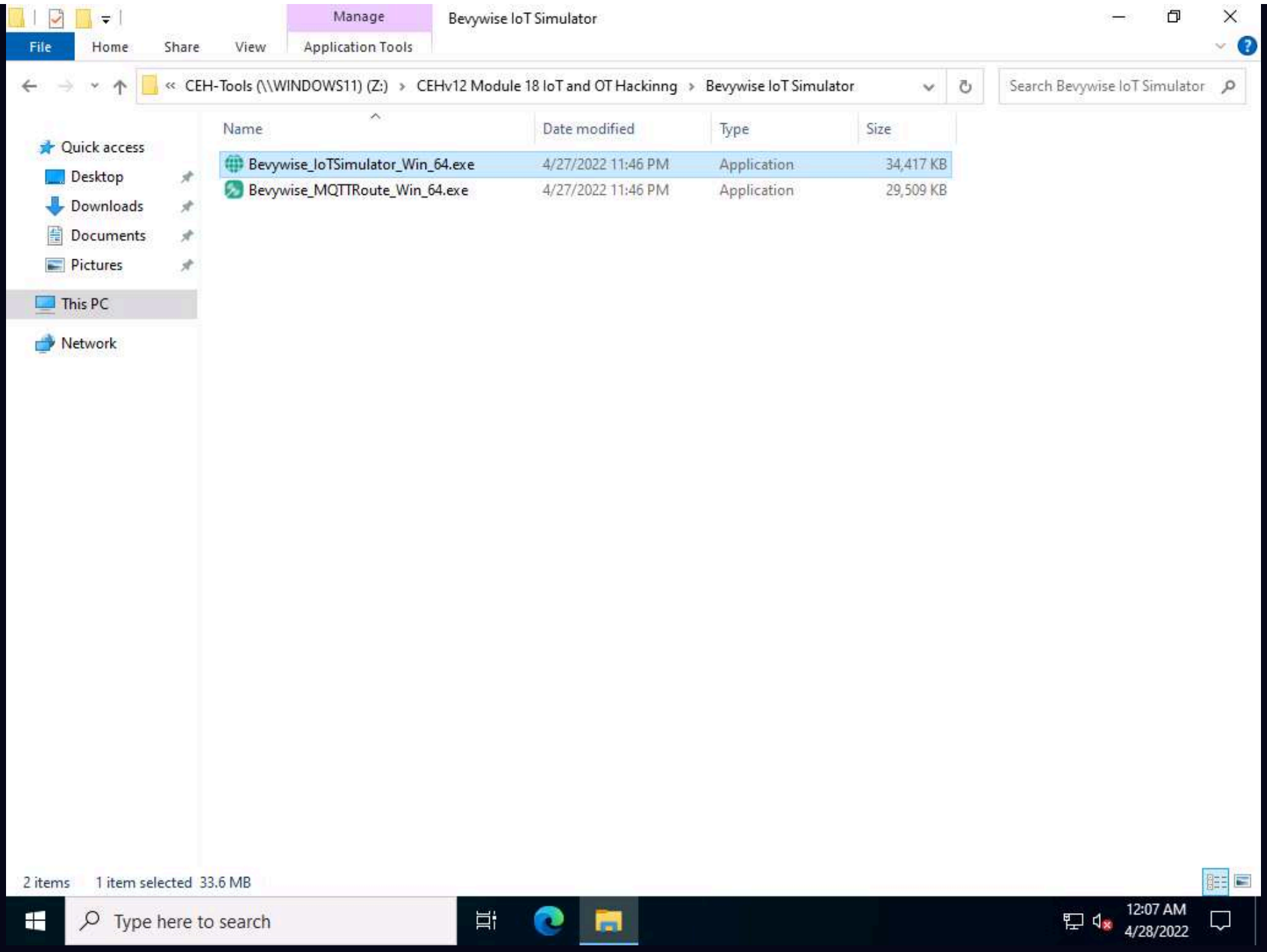


13. By default **CEH\Administrator** account is selected, type **Pa$$w0rd** in the Password field and press **Enter** to login.



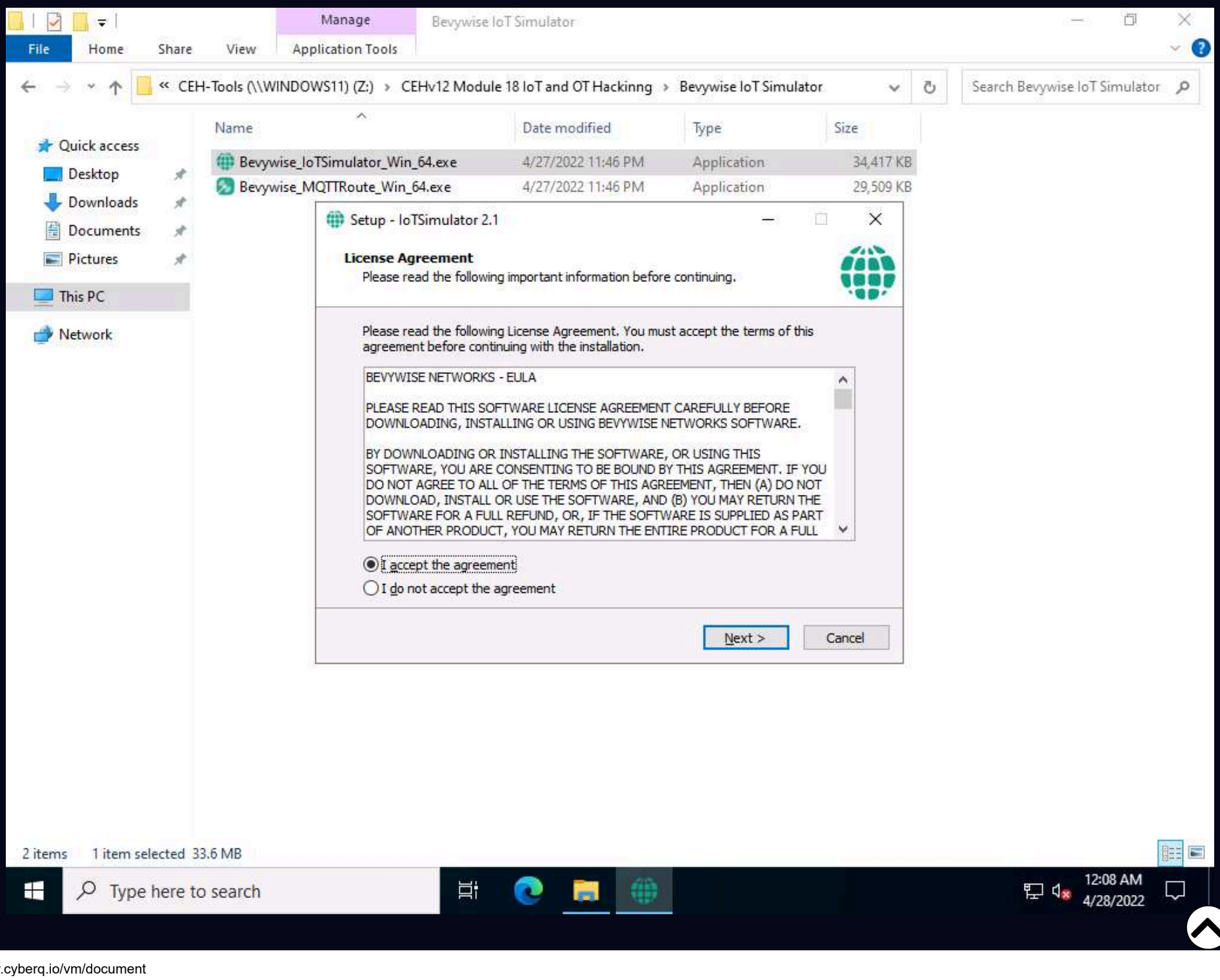
Note: If the network screen appears, click **Yes**.

14. Navigate to **Z:\CEH-Tools\CEHv12 Module 18 IoT and OT Hacking\Bevywise IoT Simulator** folder and double-click on the **Bevywise_IoTSimulator_Win_64.exe** file.



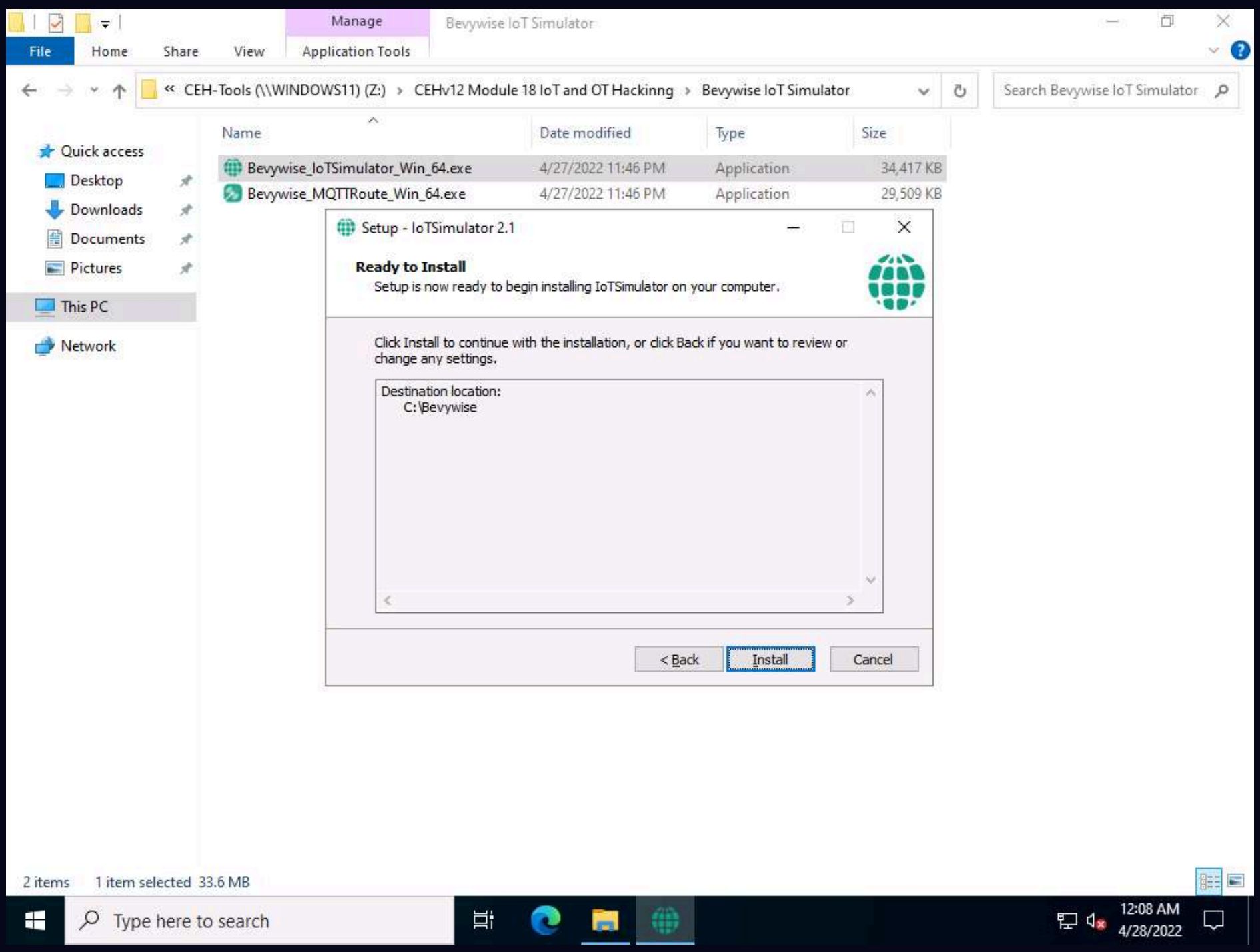
15. The **Open File - Security Warning** popup appears. Click **Run**..

16. The **Setup-IoTSimulator 2.1** setup wizard opens. Select **I accept the agreement** and click on **Next** to continue.

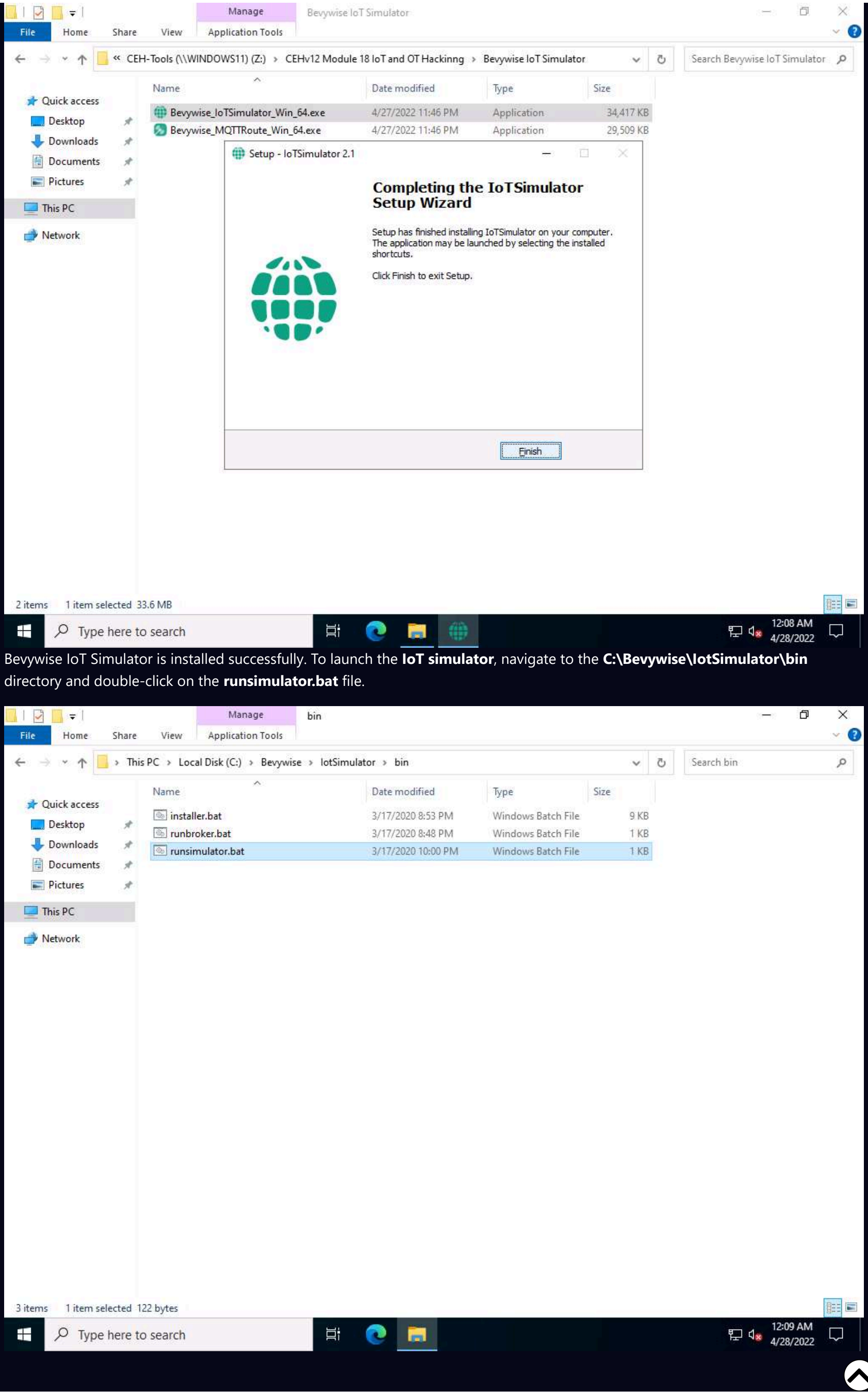


17. Keeping the default destination unchanged, click on **Next**.

18. The Ready to install screen appears, click on **Install**



19. Click on **Finish** to complete the installation.

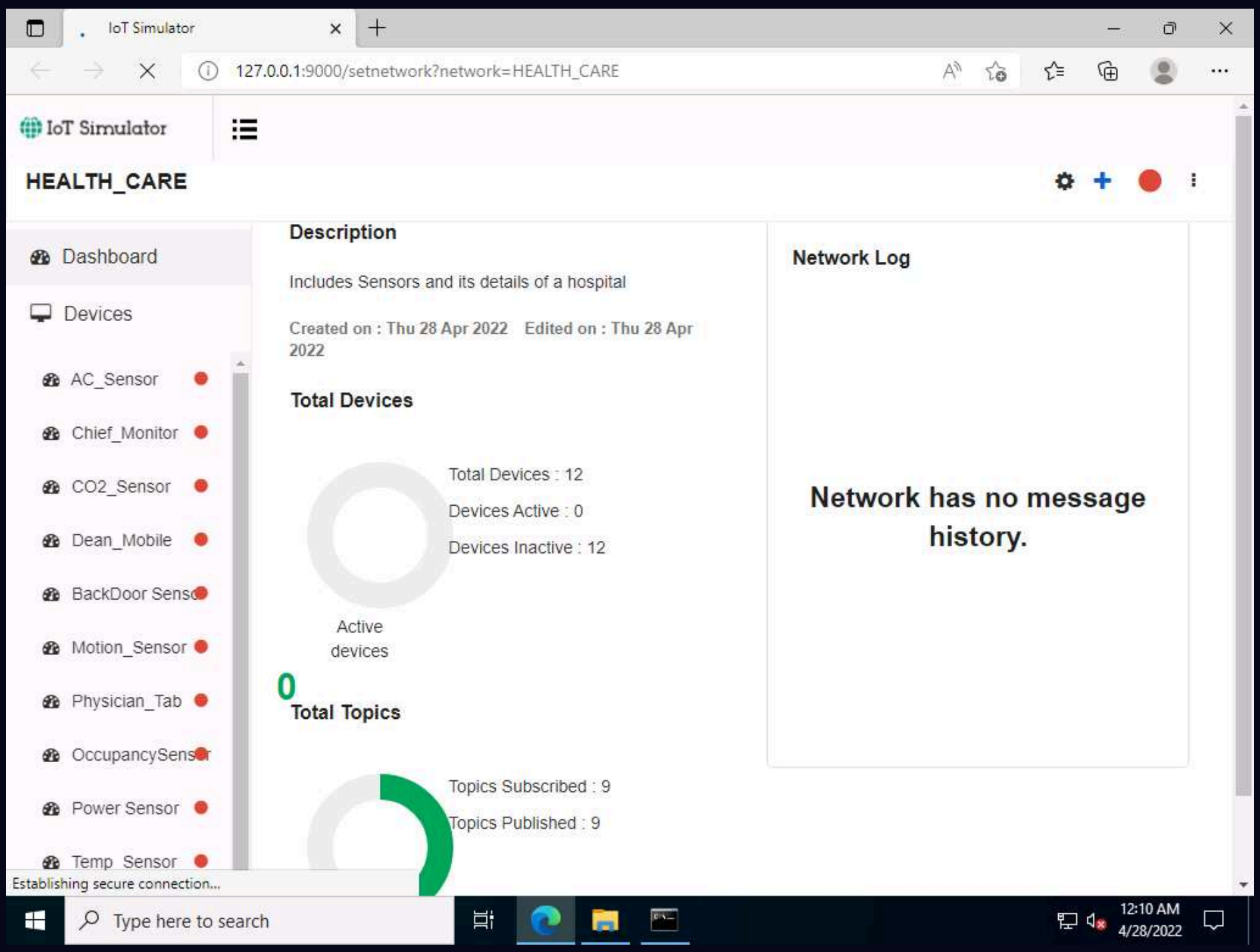


20. Bevywise IoT Simulator is installed successfully. To launch the **IoT simulator**, navigate to the **C:\Bevywise\IotSimulator\bin** directory and double-click on the **runsimulator.bat** file.

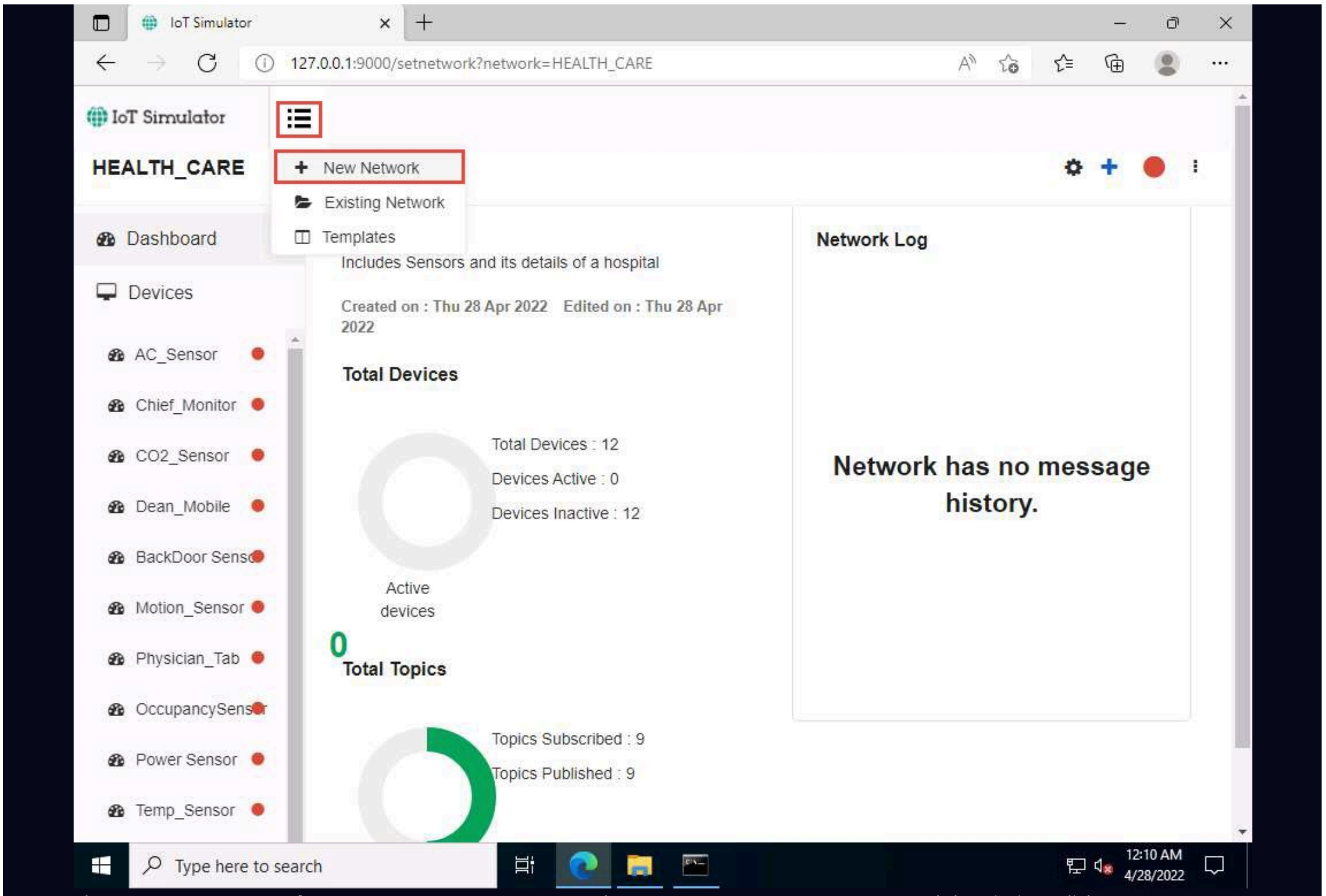
21. Upon double-clicking the **runsimulator.bat** file opens in the command prompt. If **How do you want to open this?** pop-up appears, select **Microsoft Edge** browser and click **OK** to open the URL **http://127.0.0.1:9000/setnetwork?network=HEALTH_CARE**.

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

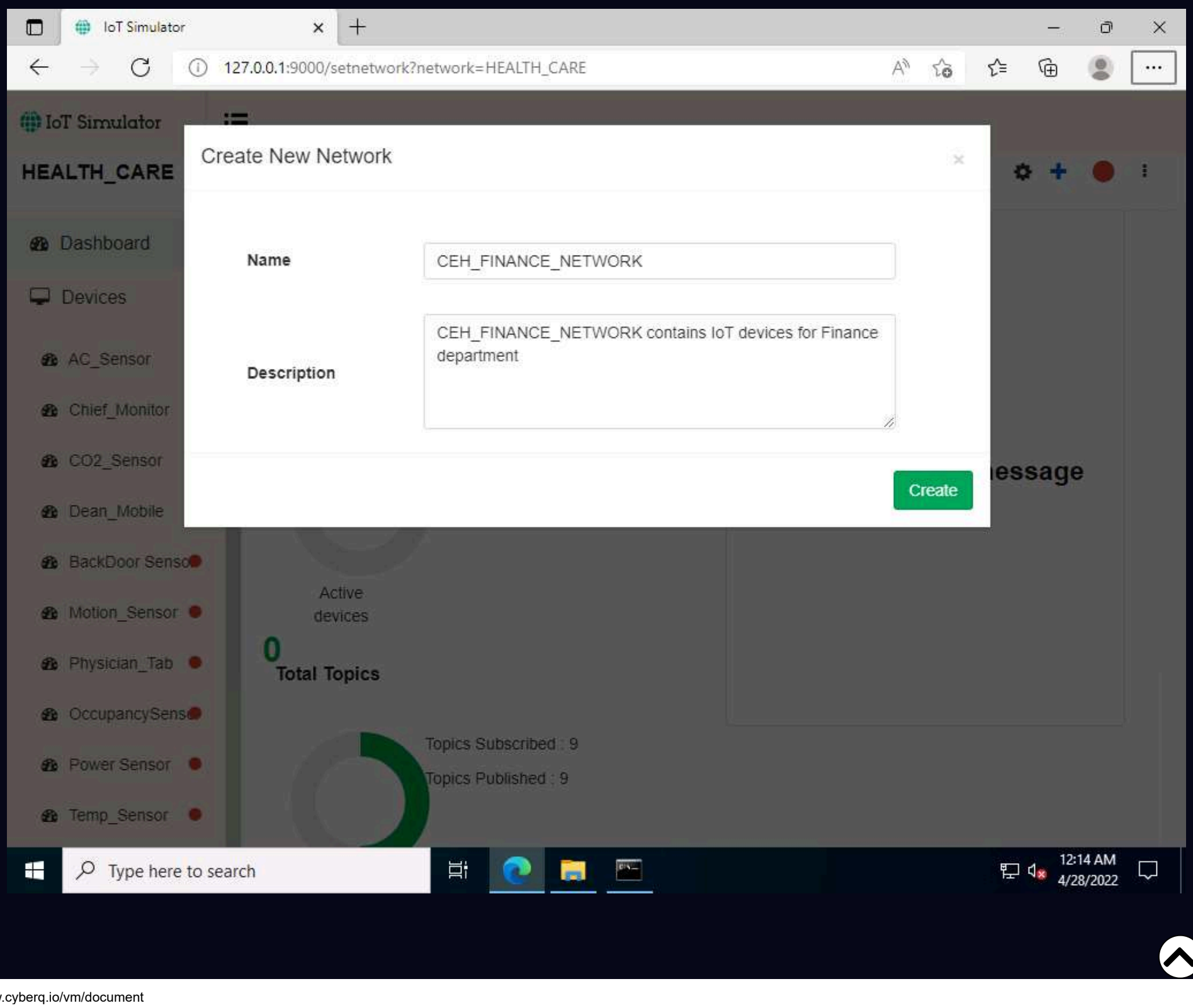
22. The web interface of the IoT Simulator opens in Edge browser. In the IoT Simulator, you can view the default network named **HEALTH_CARE** and several devices.



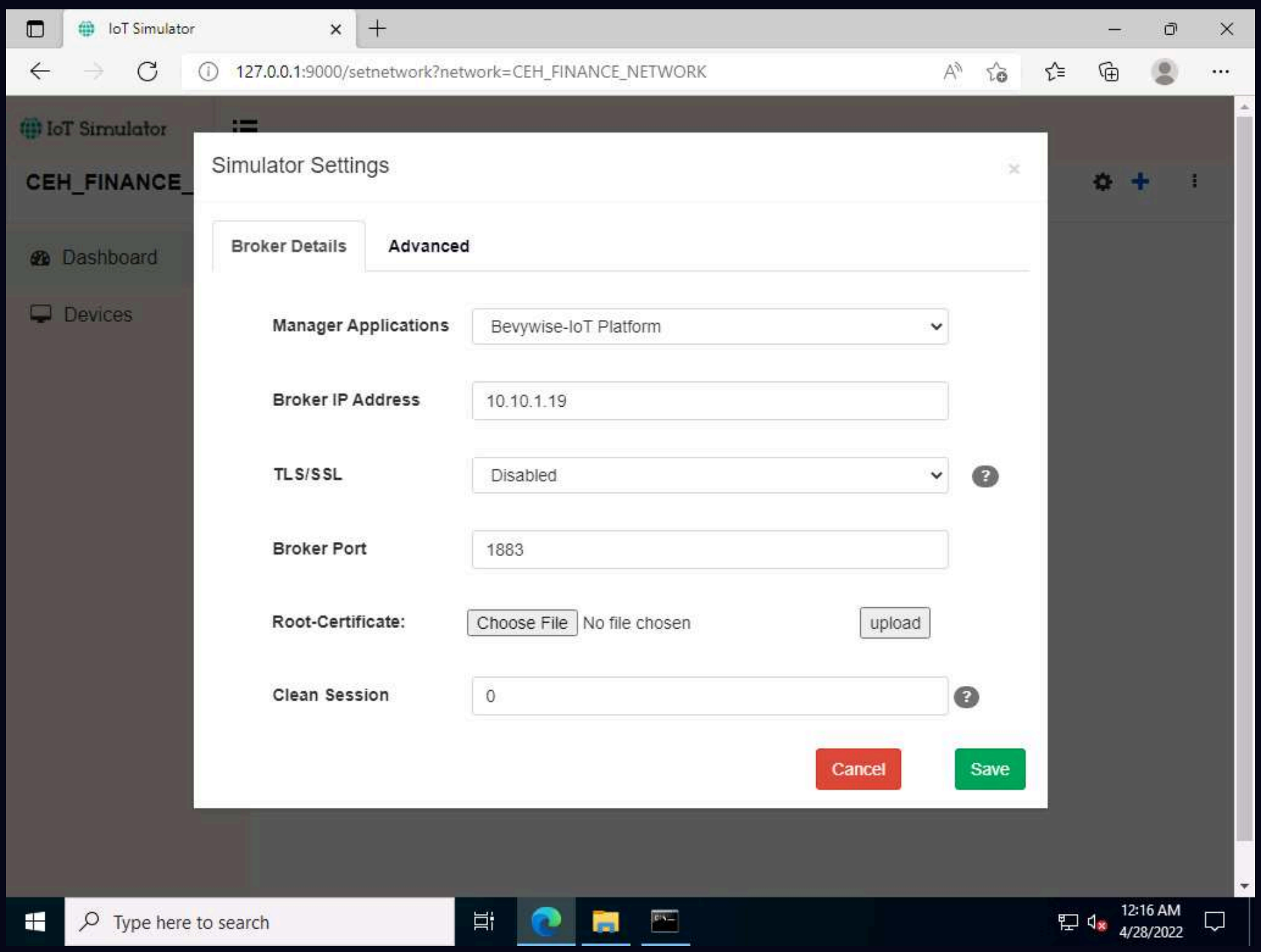
23. Next, we will create a **virtual IoT network** and **virtual IoT devices**. Click on the **menu** icon and select the **+New Network** option.



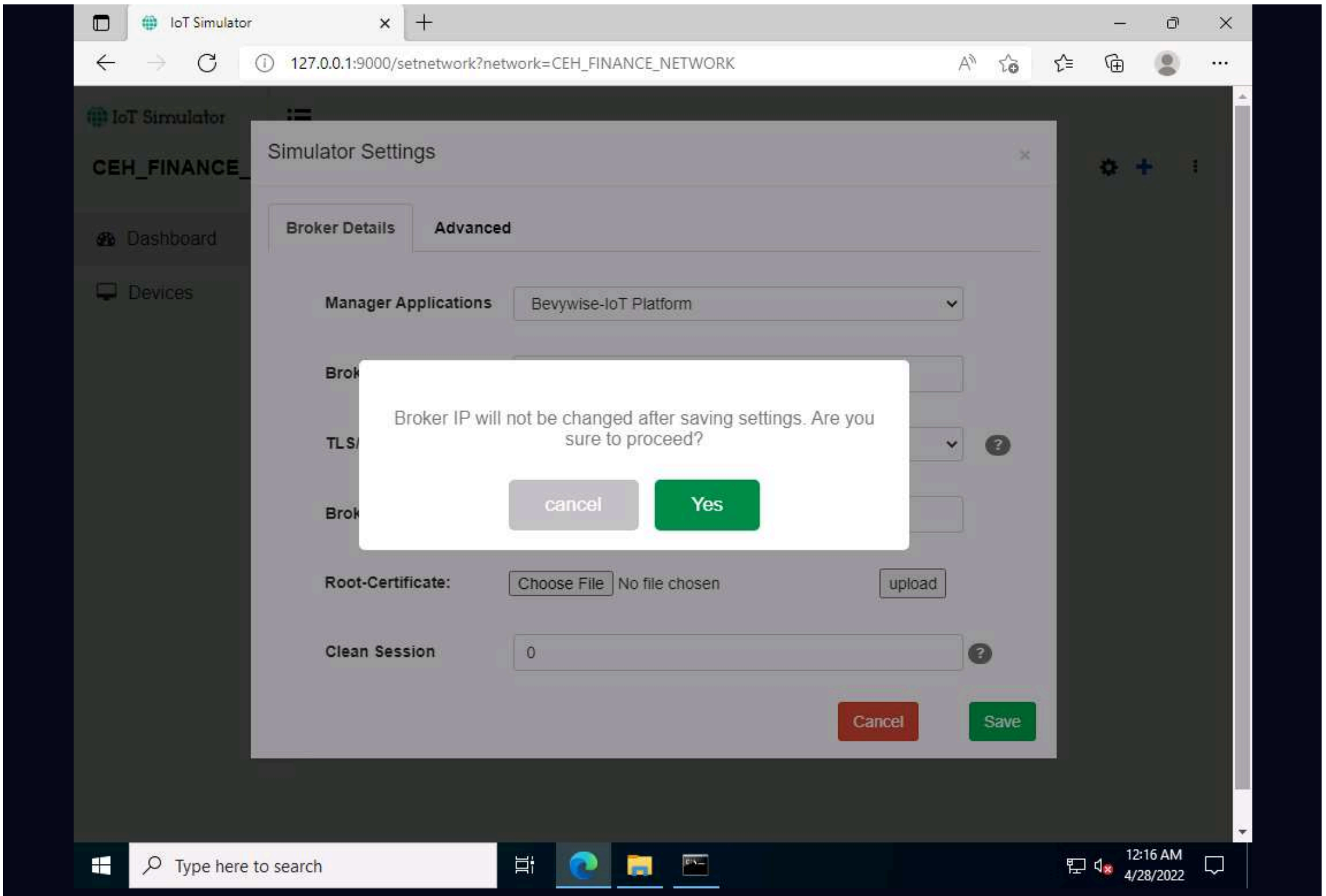
24. The **Create New Network** popup appears. Type any name (here, **CEH_FINANCE_NETWORK**) and description. Click on **Create**.



25. 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**.

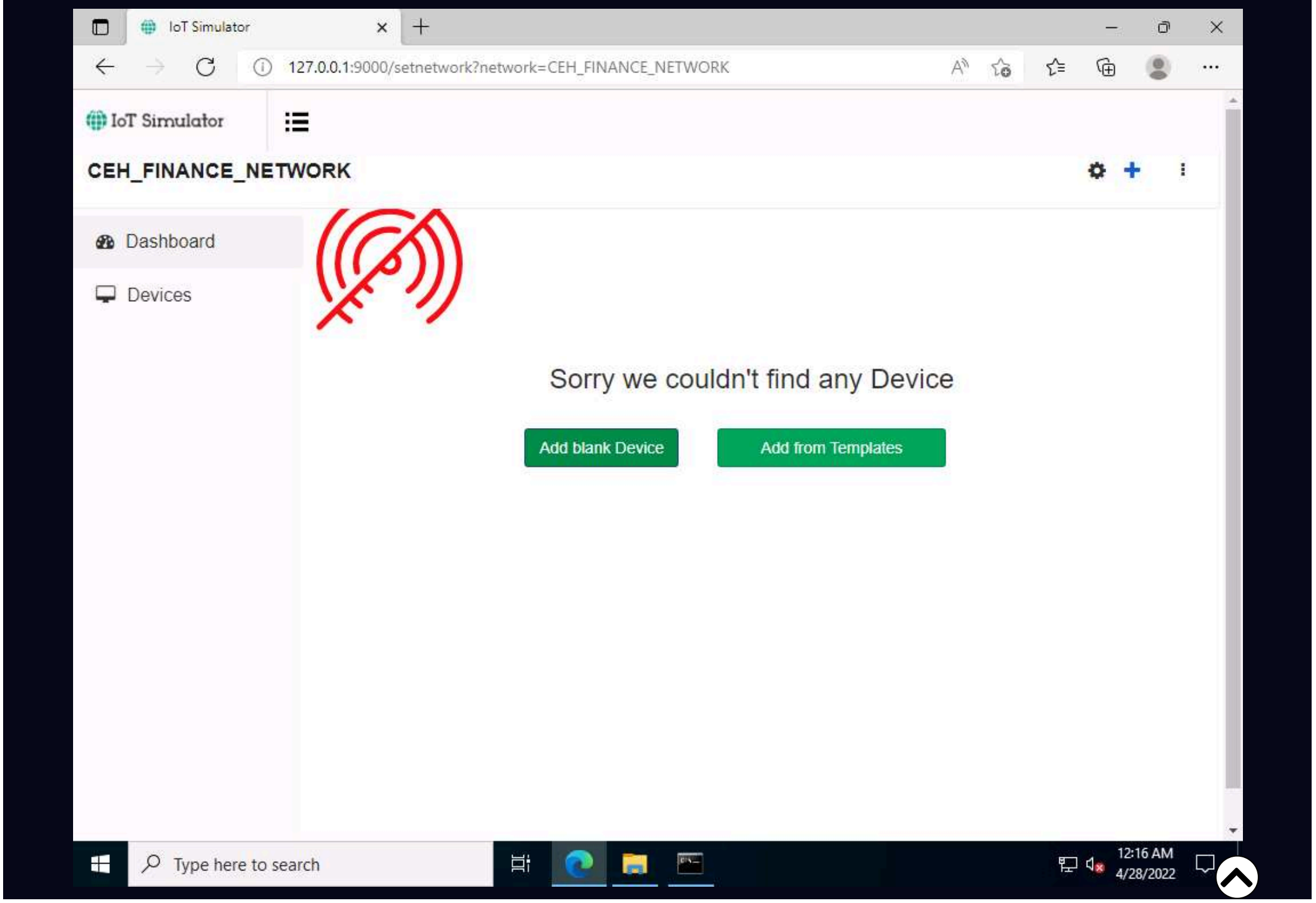


26. To proceed with the network creation, click on **Yes**.

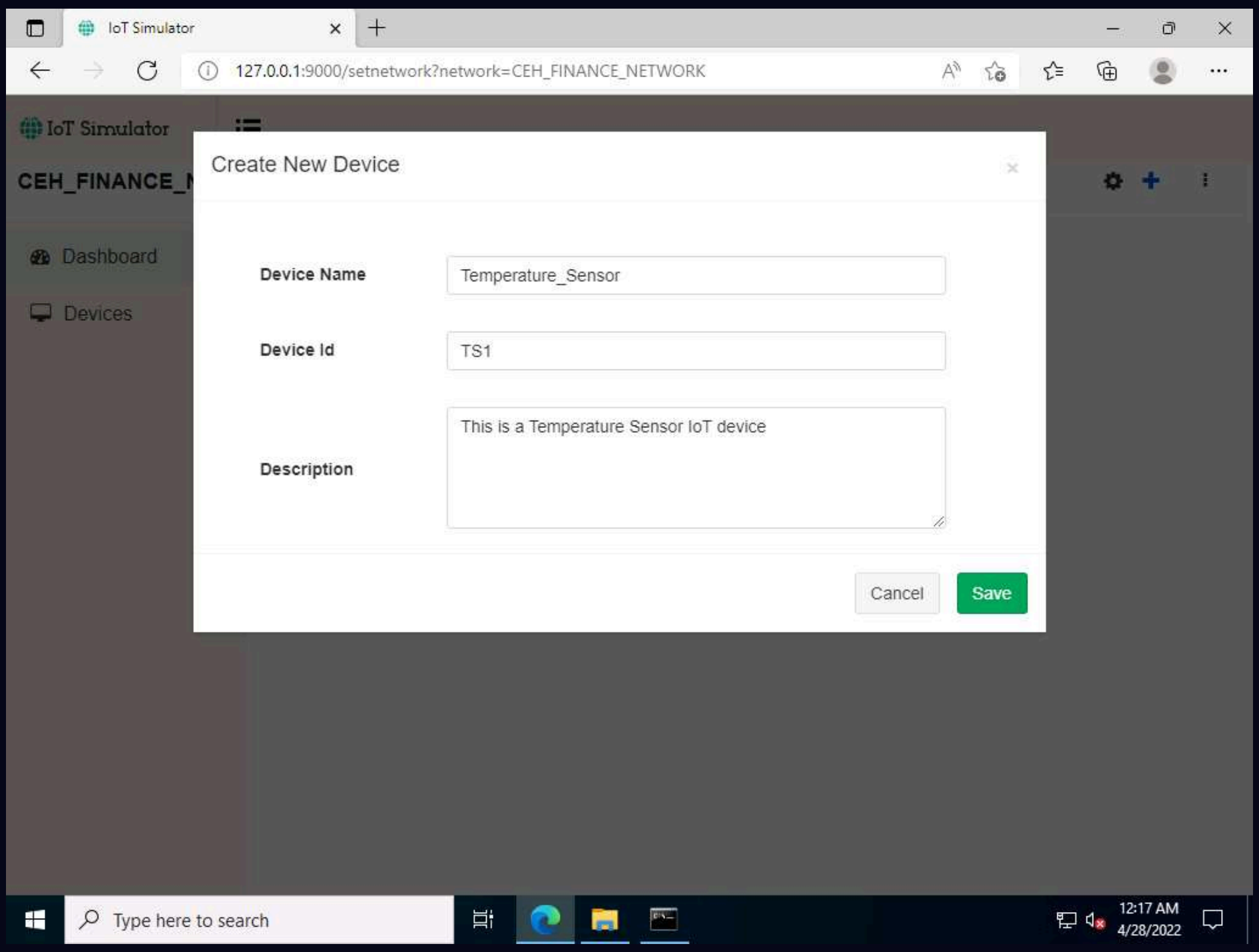


Note: If **Configuration Saved** pop-up appears. Click on **OK** to continue. This step completes the creation of the virtual IoT network.

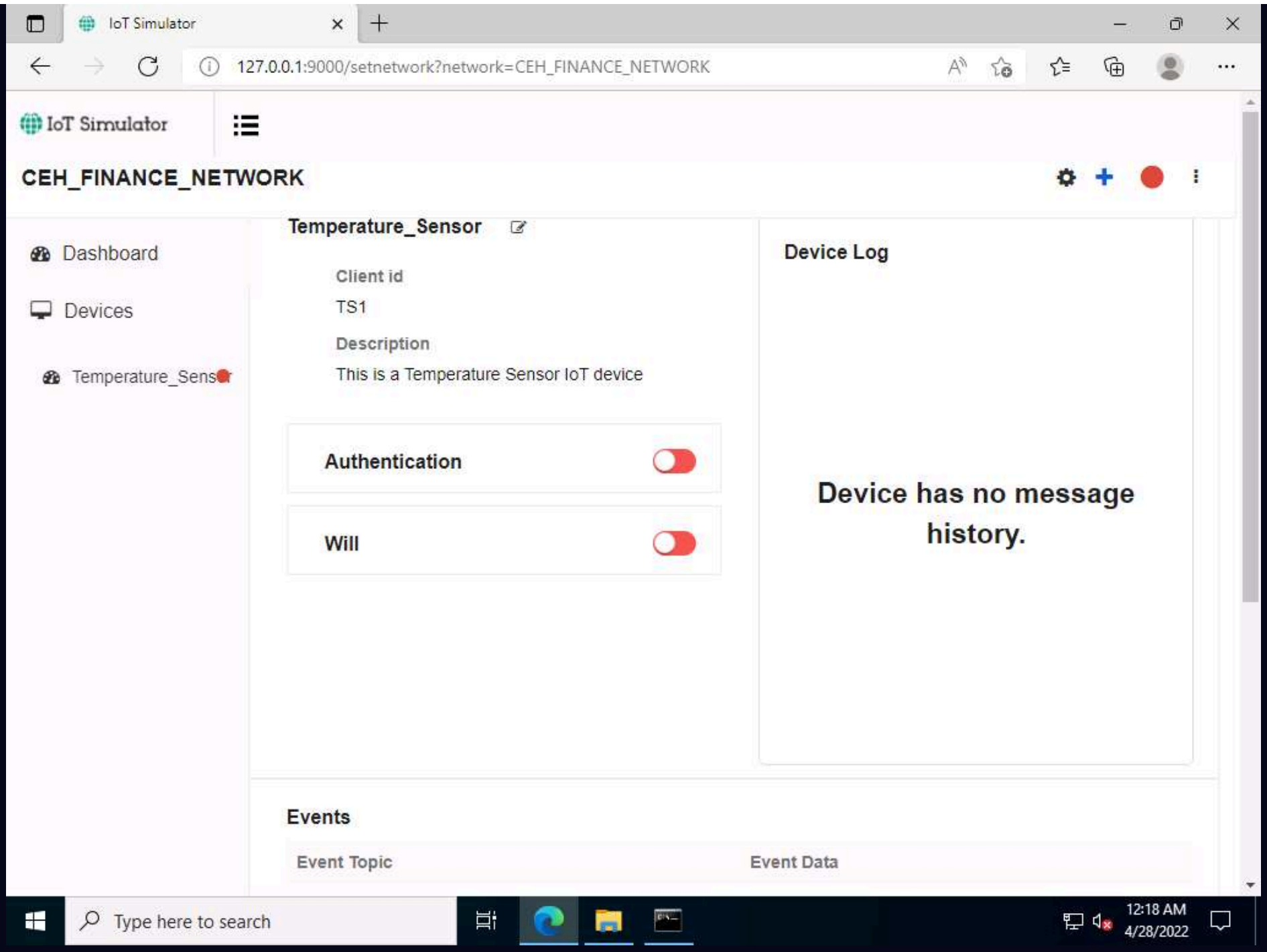
27. To add IoT devices to the created network, click on the **Add blank Device** button.



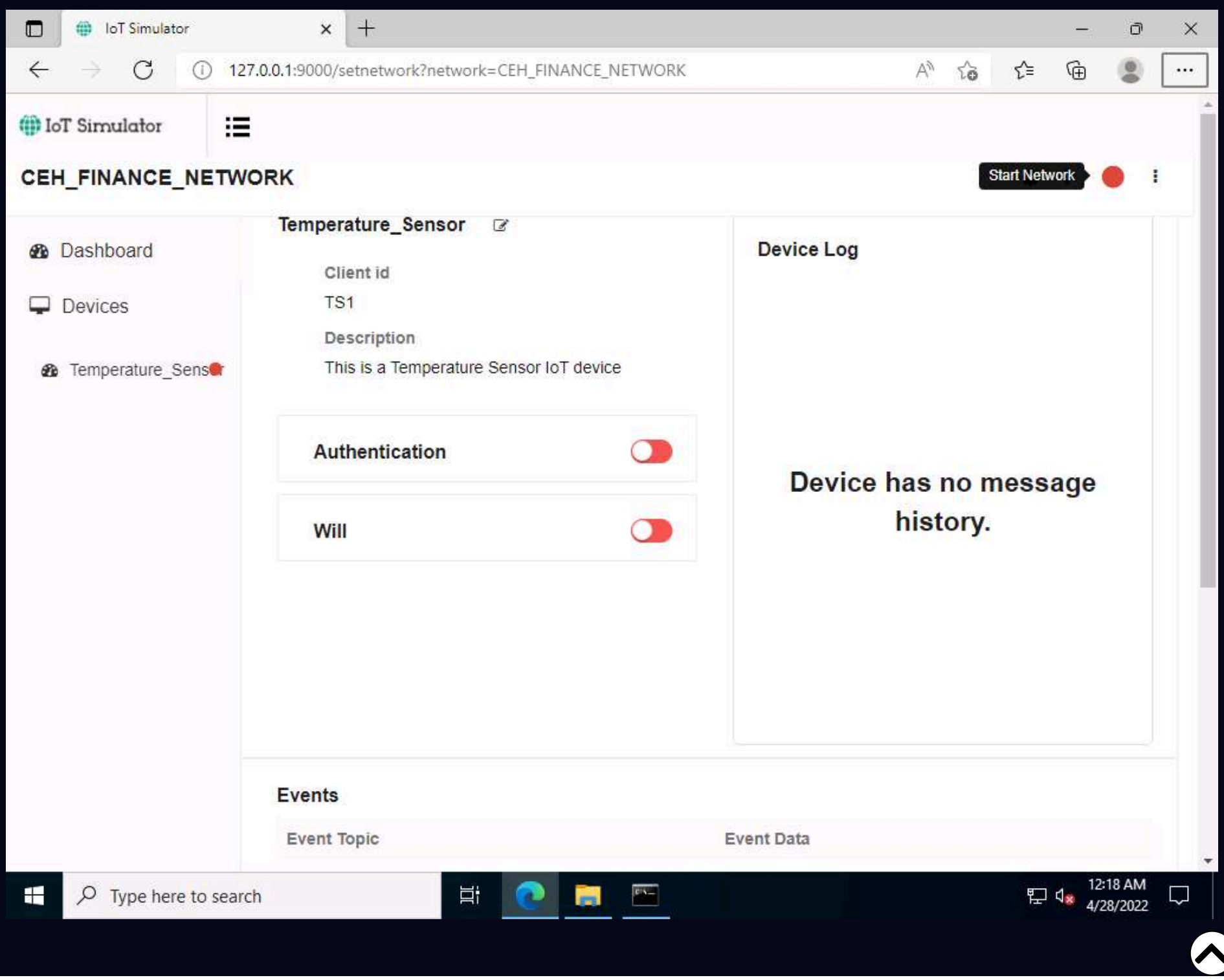
28. The **Create New Device** popup opens. Type the device name (here, we use **Temperature_Sensor**), enter Device Id (here, we use **TS1**), provide a **Description** and click on **Save**.



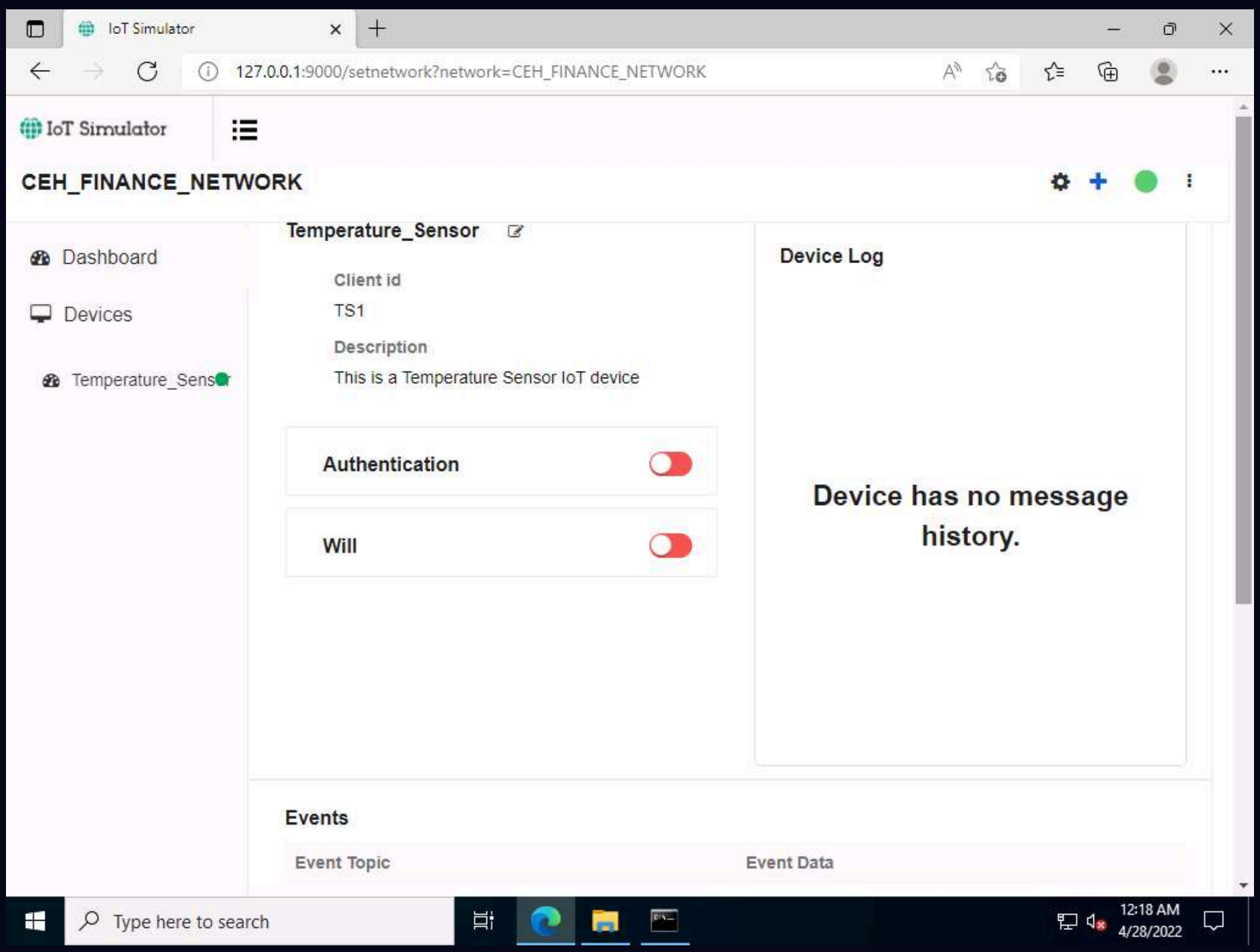
29. The device will be added to the **CEH_FINANCE_NETWORK**.



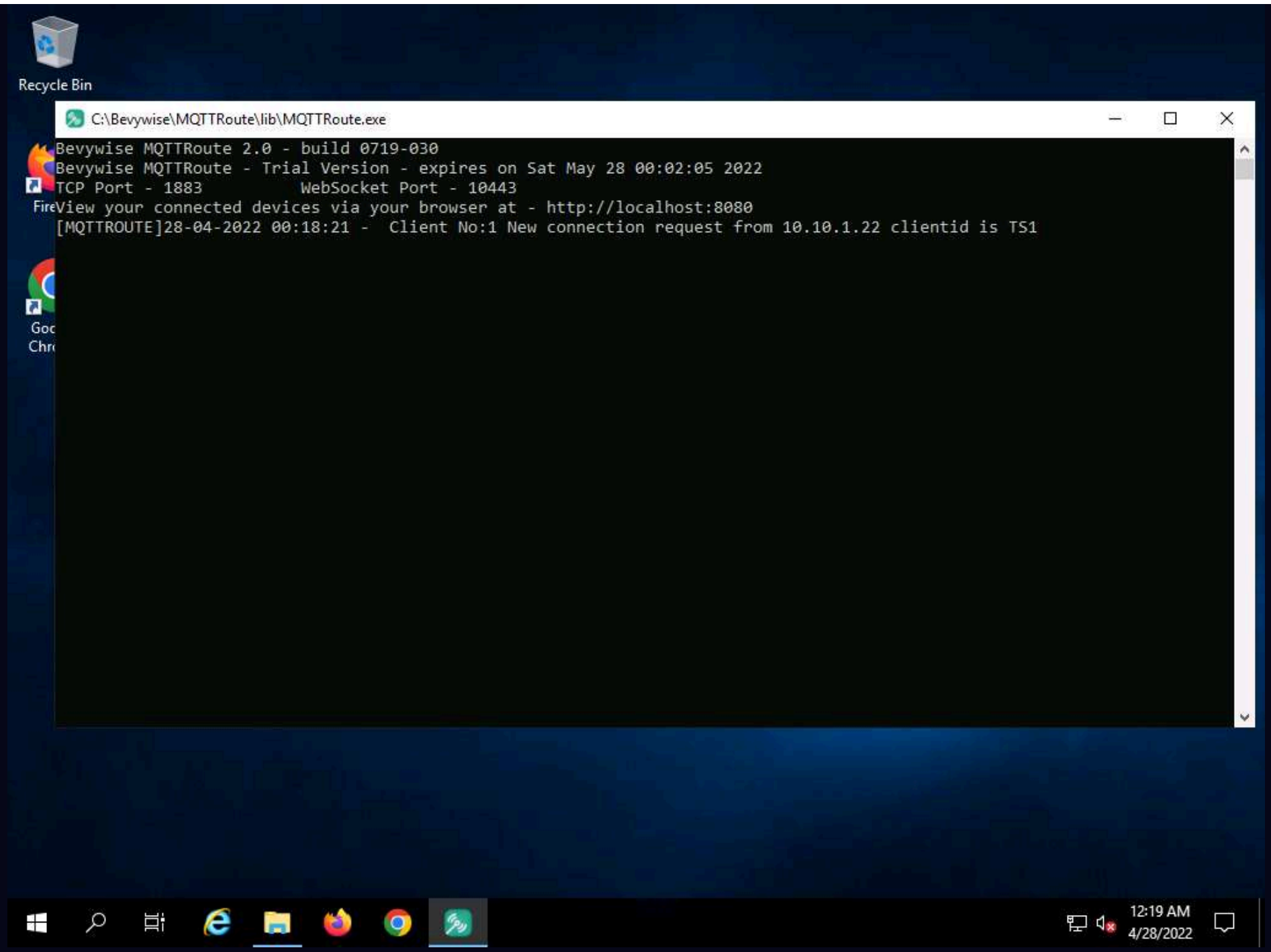
30. To connect the Network and the added devices to the server or Broker, click on the **Start Network** red color circular icon in right corner.



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



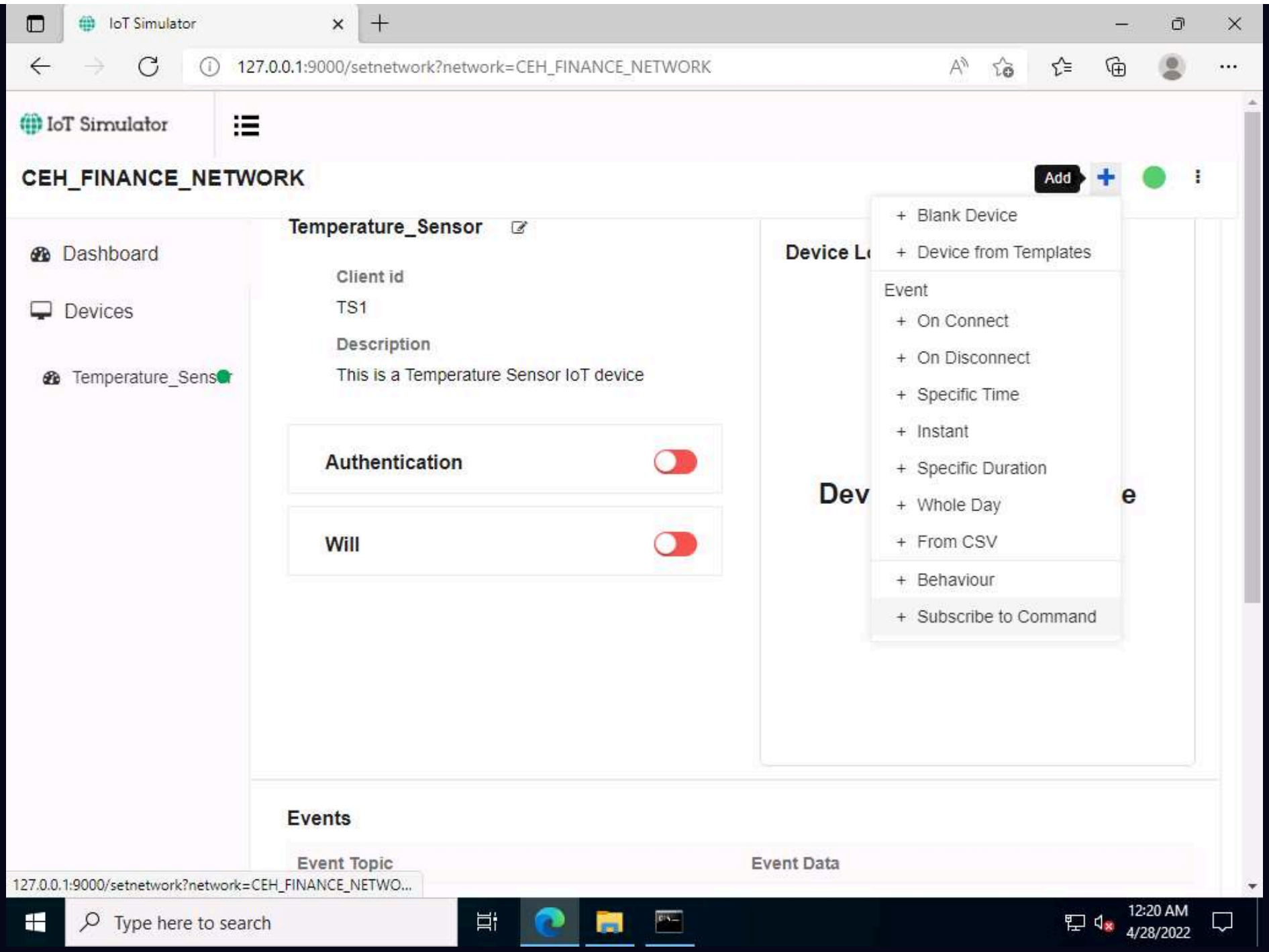
32. Next, switch to the **CEHv12 Windows Server 2019** machine. Since the Broker was **left running**, you can see a connection request from machine **10.10.1.22** for the device **TS1**.



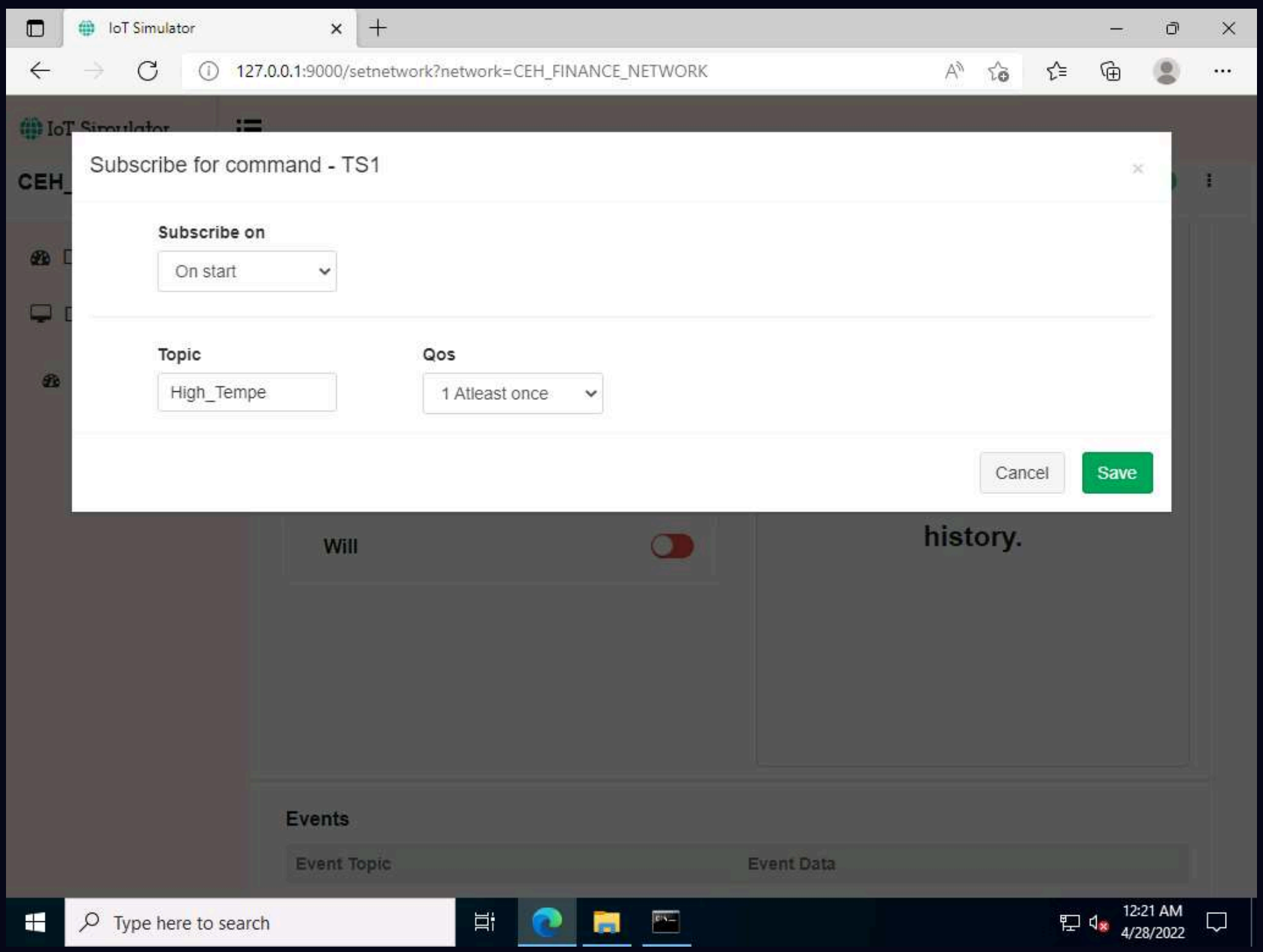
33. Switch back to **CEHv12 Windows Server 2022** machine.

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

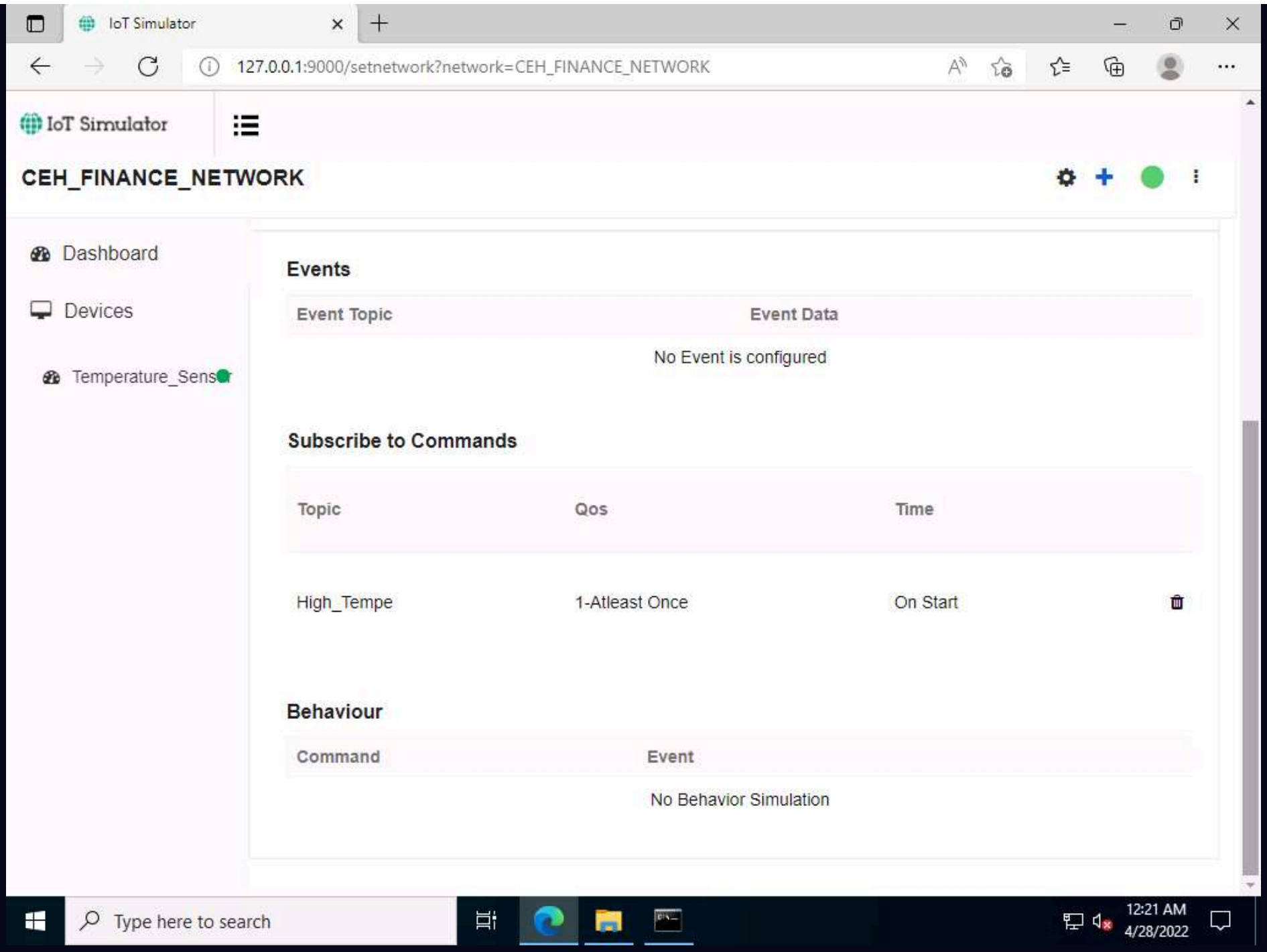
35. Click on the **Plus** icon in **the top right corner** and select the **Subscribe to Command** option.



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

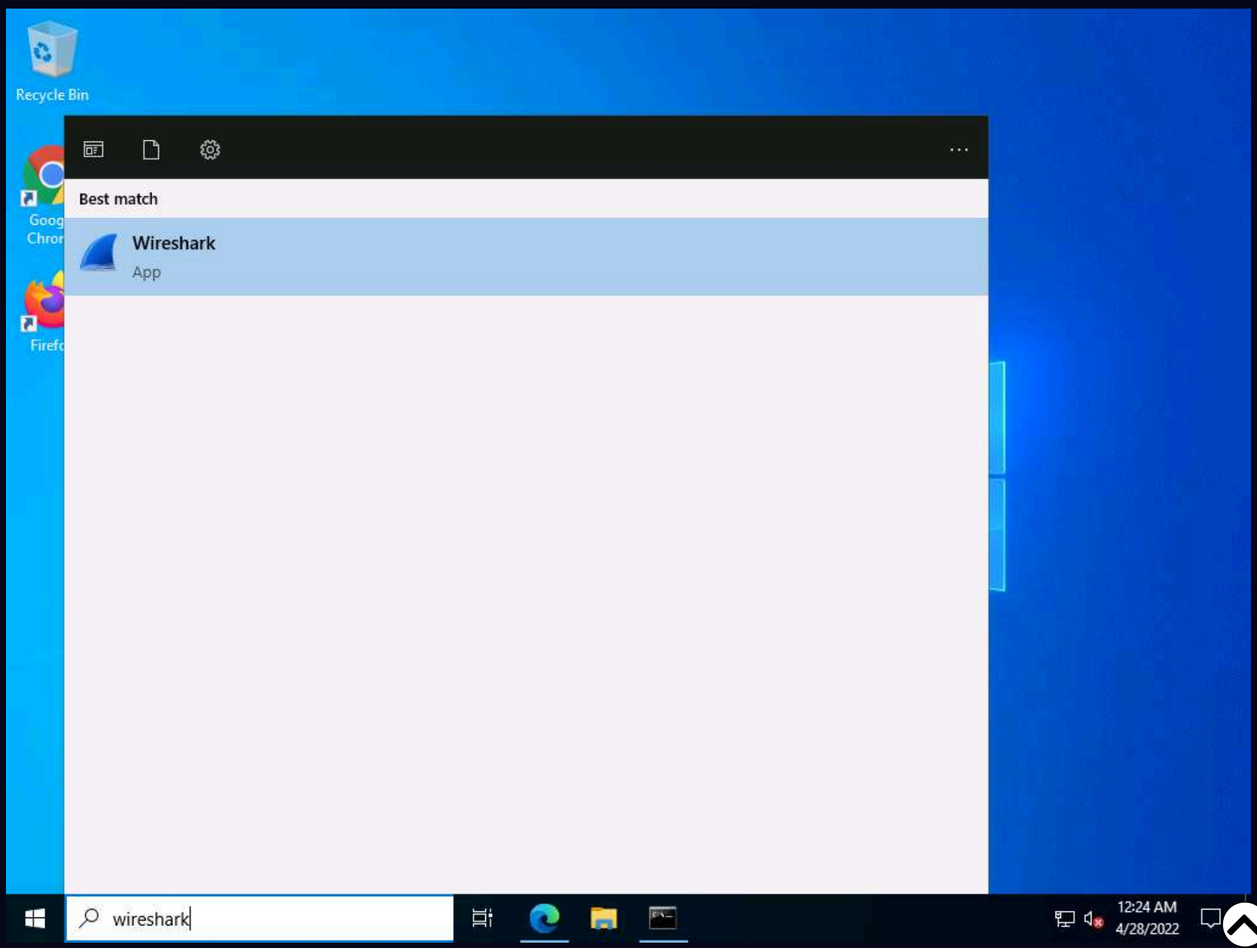


37. Scroll down the page, you can see the **Topic** added under the **Subscribe to Commands** section.



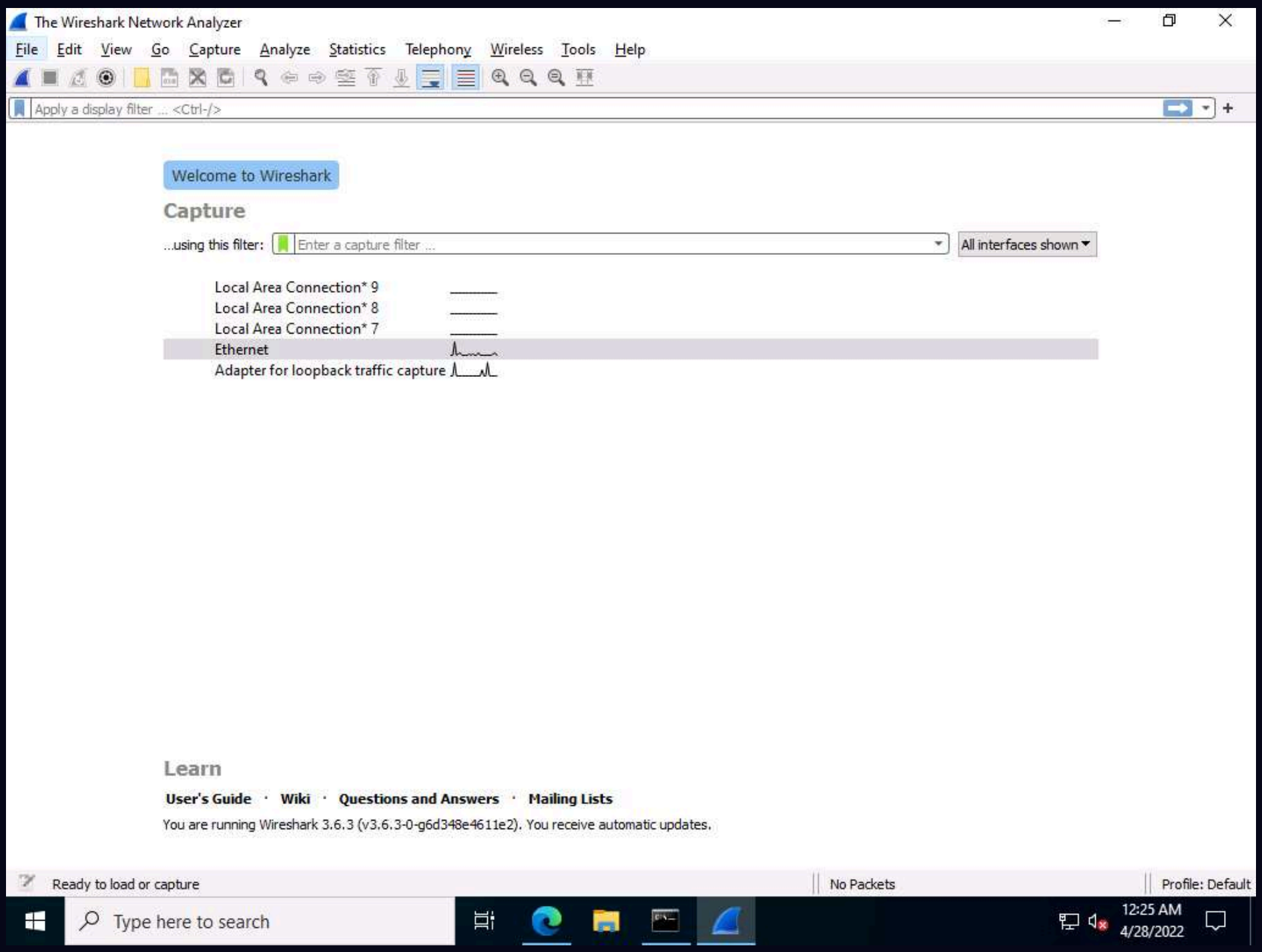
38. Next, we will capture the traffic between the **virtual IoT network and the MQTT Broker** to monitor the secure communication.

39. Minimise the Edge browser. Click on **Type here to search** at the bottom left of the desktop, type wireshark and select Wireshark from the results to launch the **Wireshark** from the application list.



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

> Note: Make sure you have selected interface which has **10.10.1.22** as the IP address.
>
> Note: If Software update popup appears click on **Skip this version**.

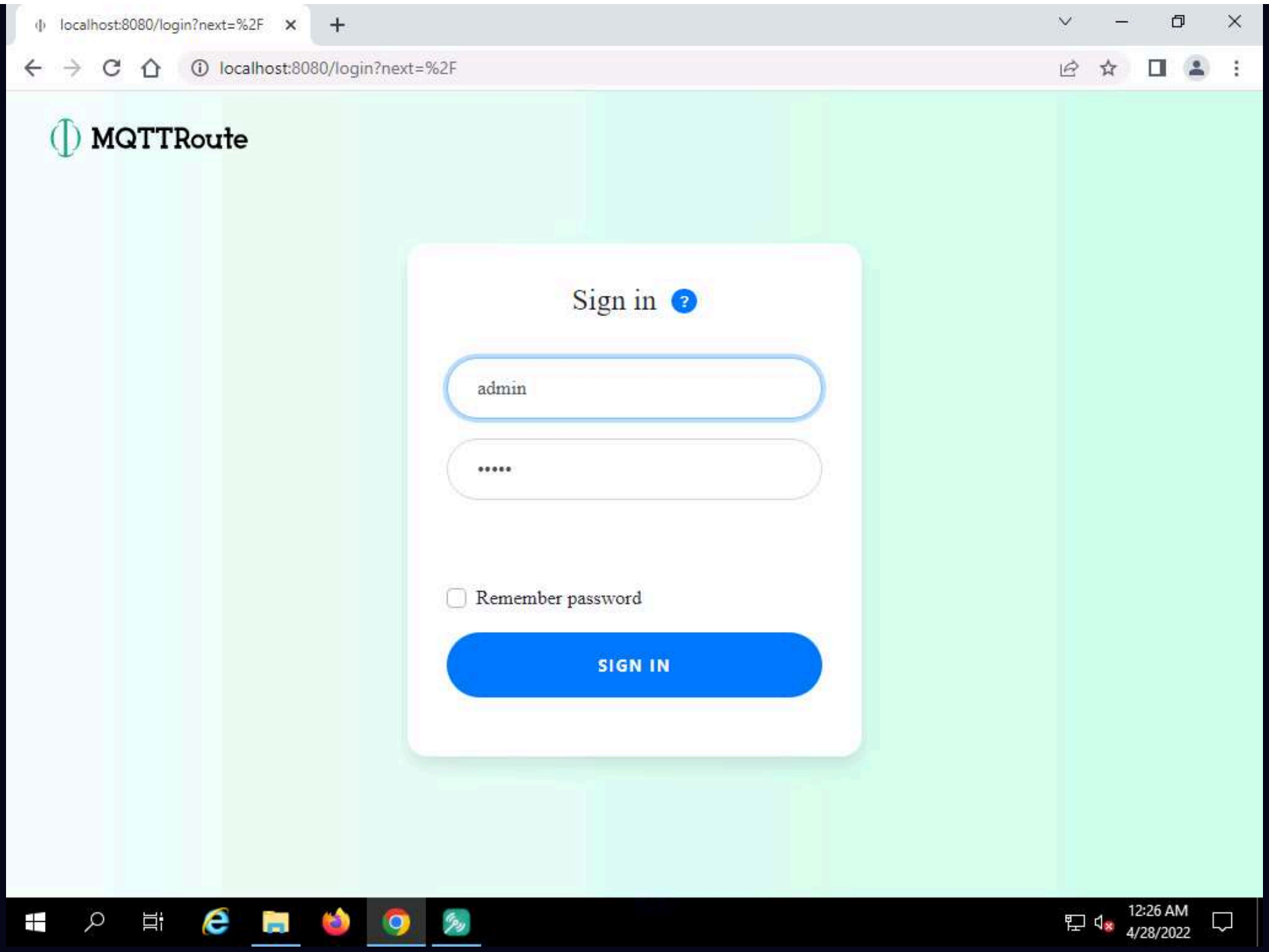


41. Click on the **Start Wireshark** icon to start the capturing packets, leave the Wireshark running.

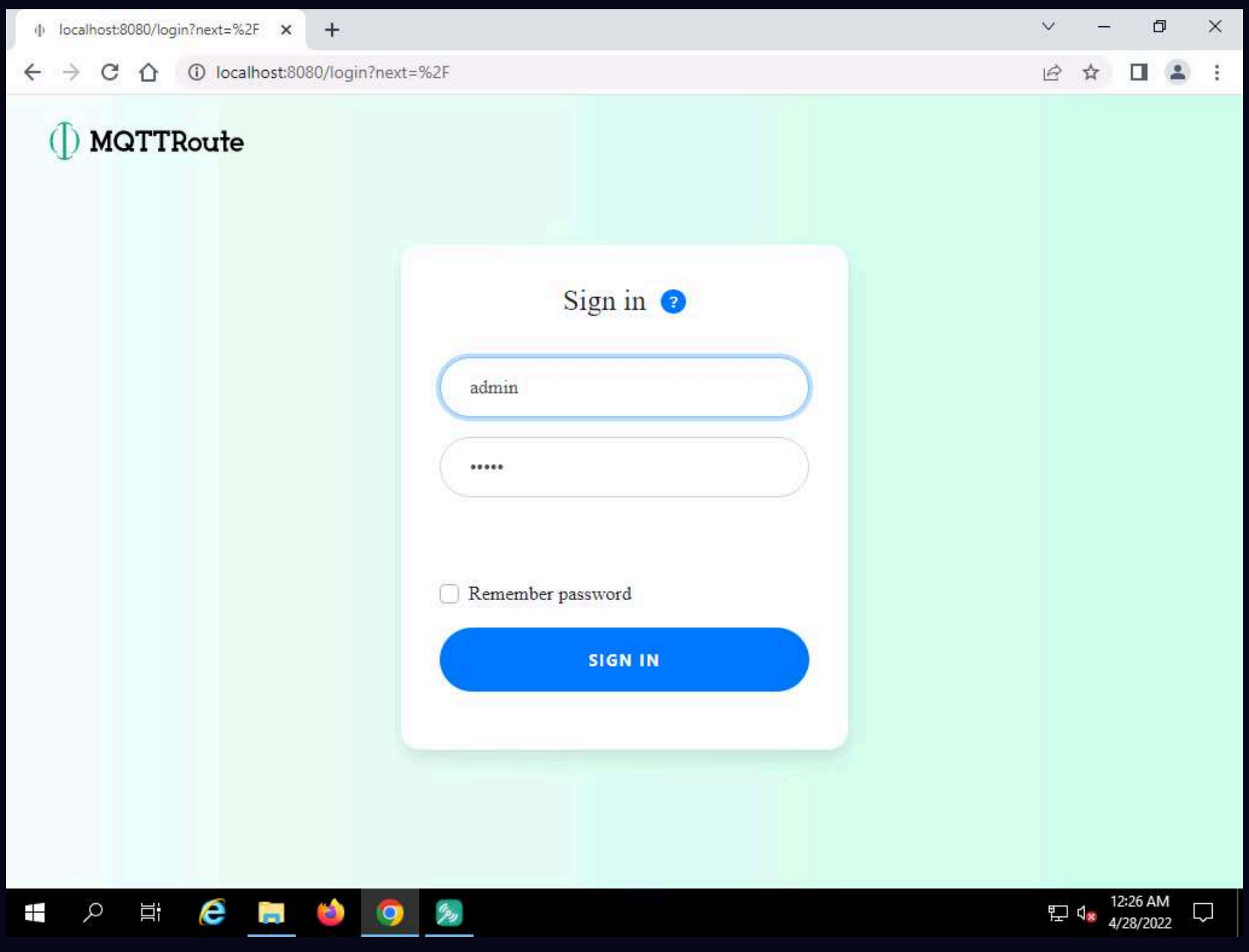
42. Leave the IoT simulator running and switch to the **CEHv12 Windows Server 2019** machine.

43. Minimise all opened applications and windows, Open Chrome browser, type **http://localhost:8080** and press **Enter**.

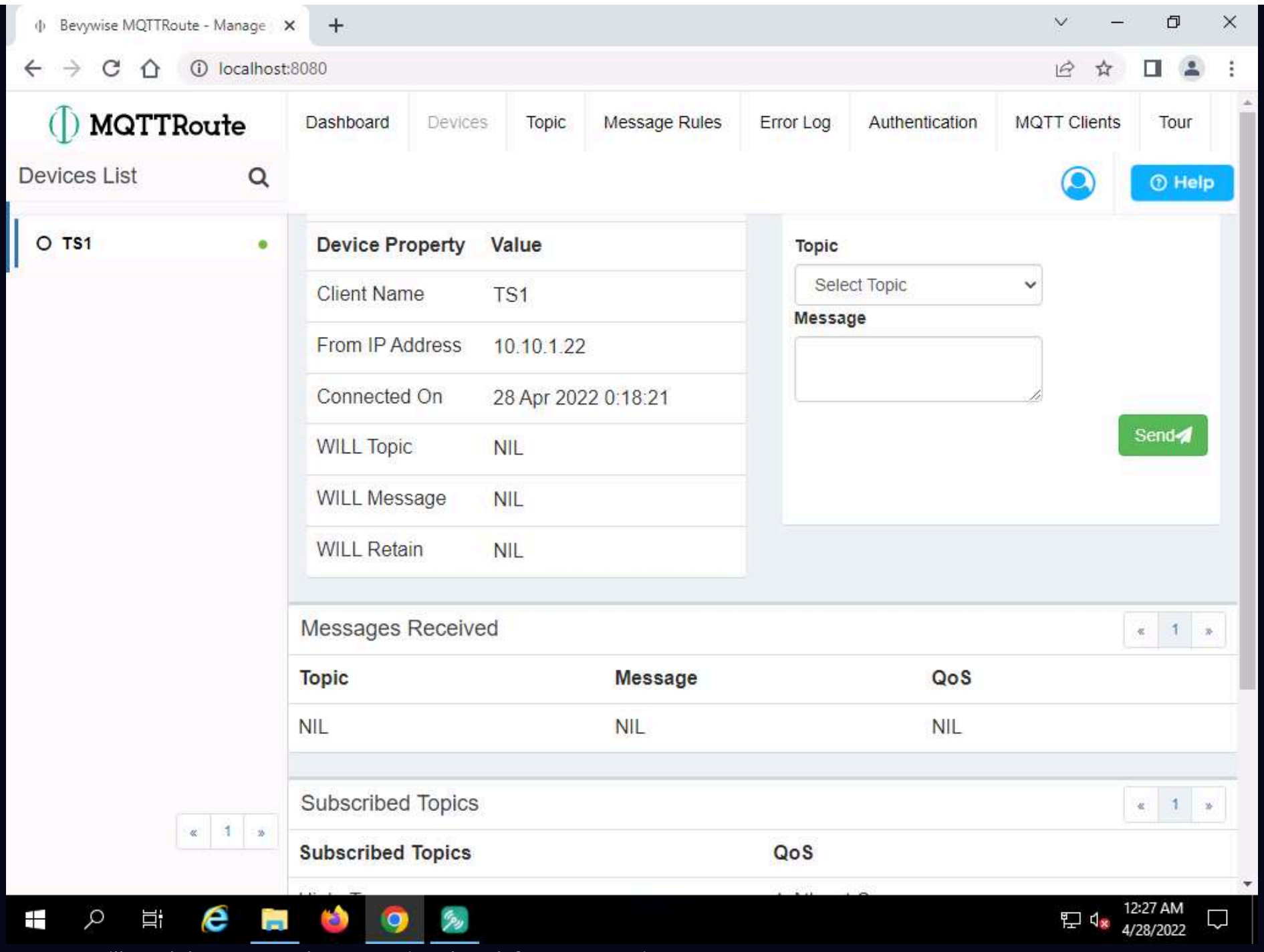
> Note: Do not use Internet Explorer web browser to open the above URL.



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



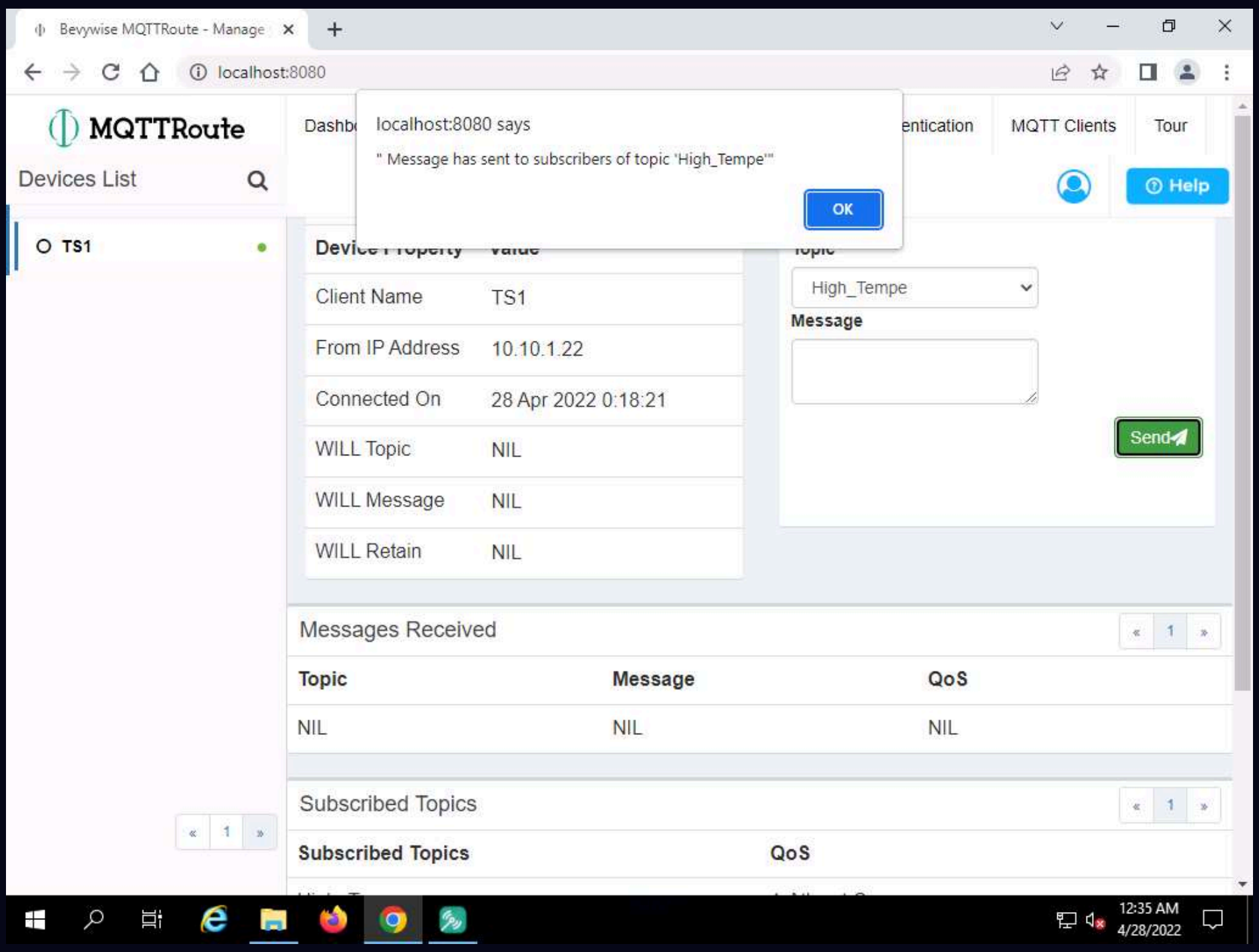
45. Navigate to **Devices** menu. You will be able to see the connected device **TS1** in the left pane.



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

47. Go to the **Command Send** section, select **Topic** as **High_Tempe**, type **Alert for High Temperature** and click on the **Send** button.

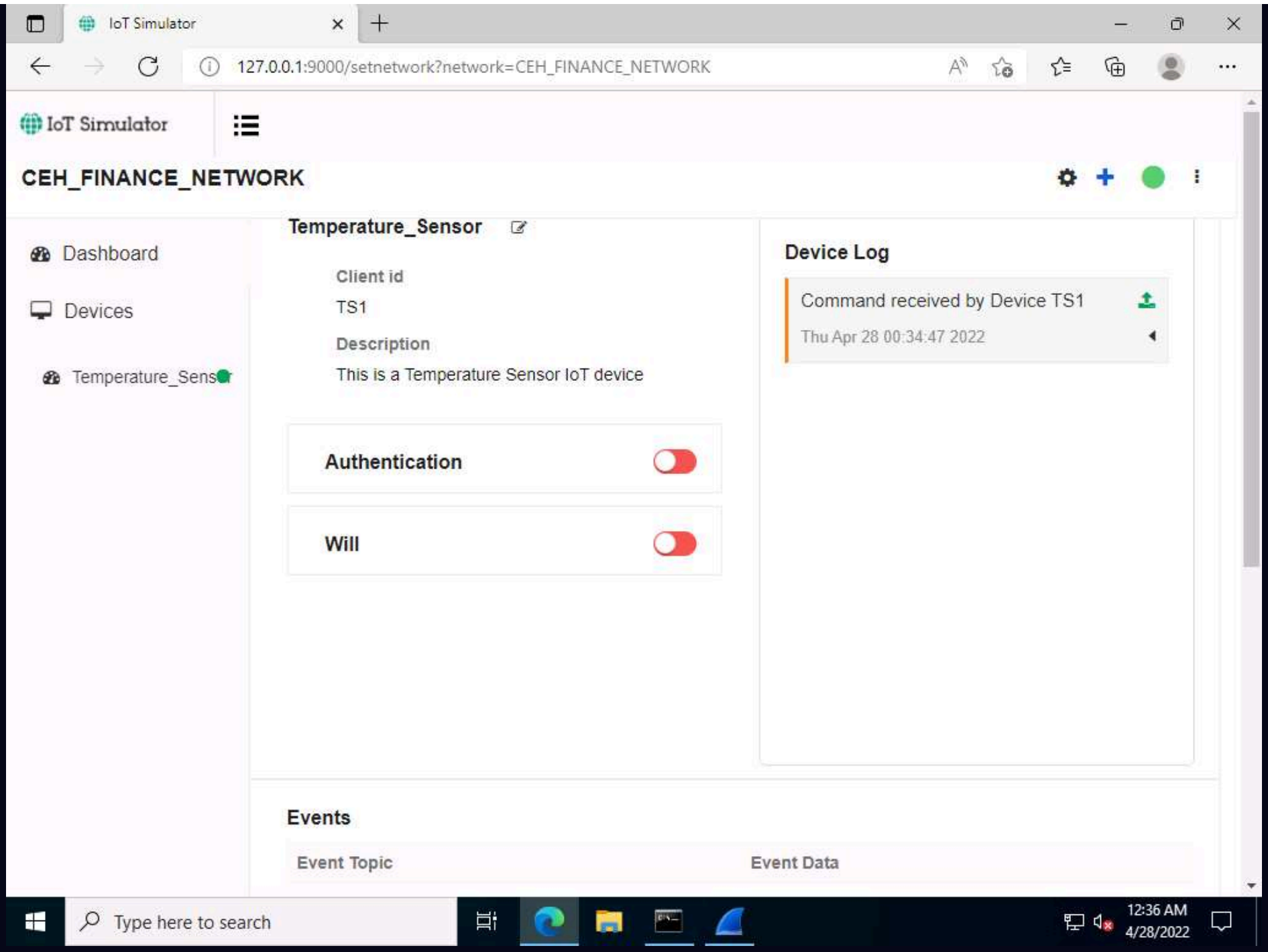
48. The alert popup appears, then click on **OK**.



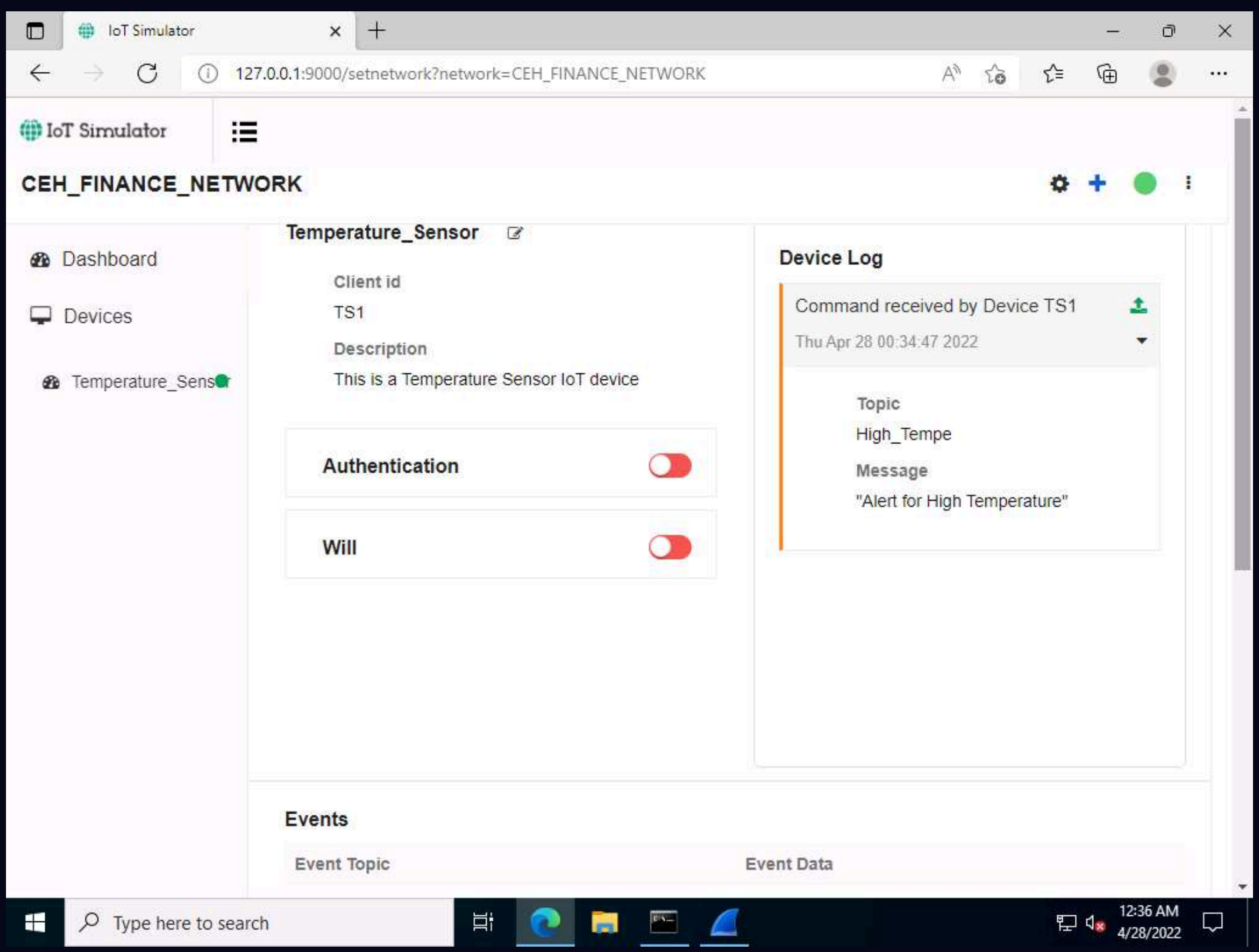
49. The message has been sent to the device using this topic.

50. Next, switch to **CEHv12 Windows Server 2022** machine.

51. 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.

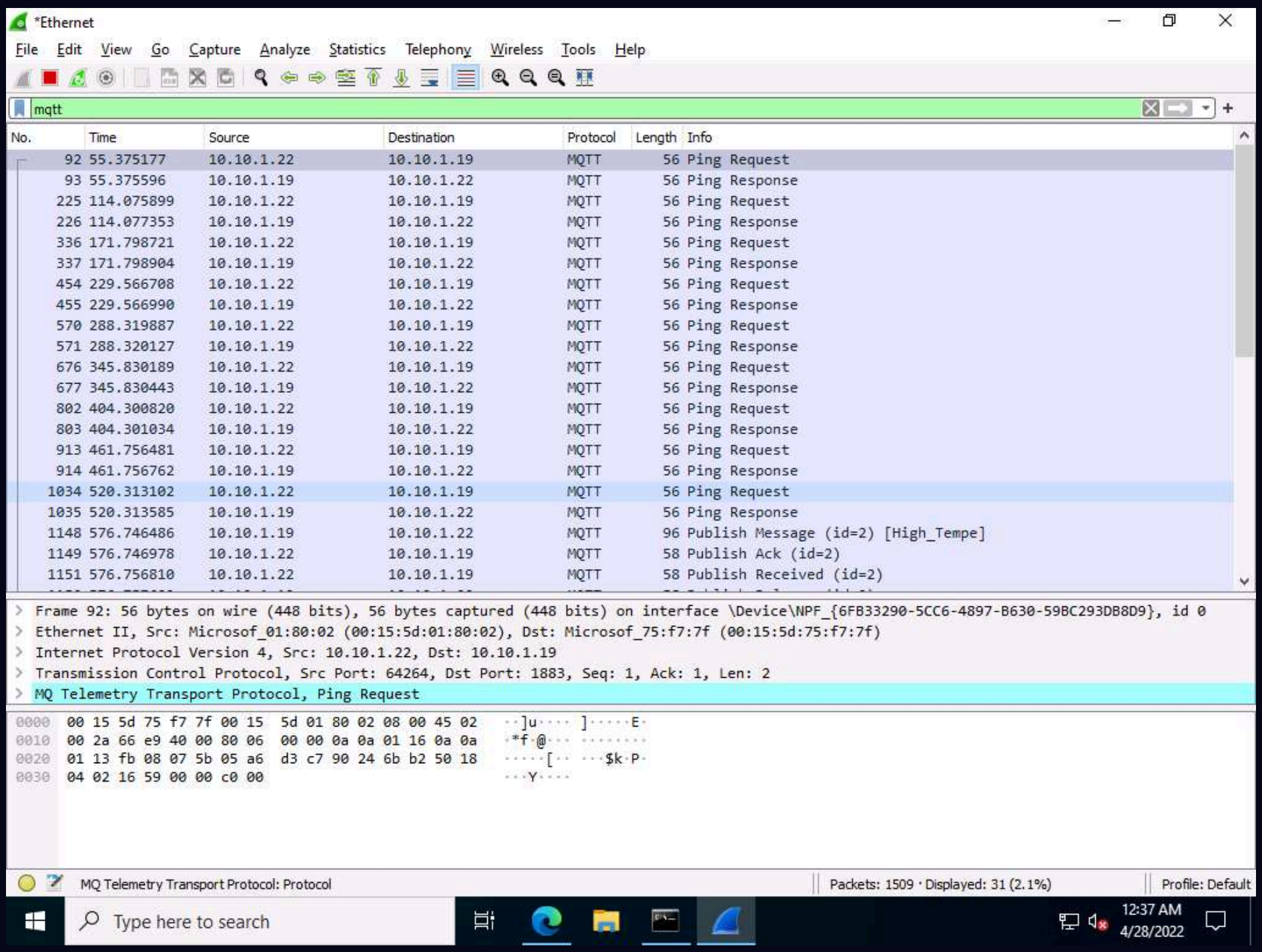


52. You can see the alert message "**Alert for High Temperature**"



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

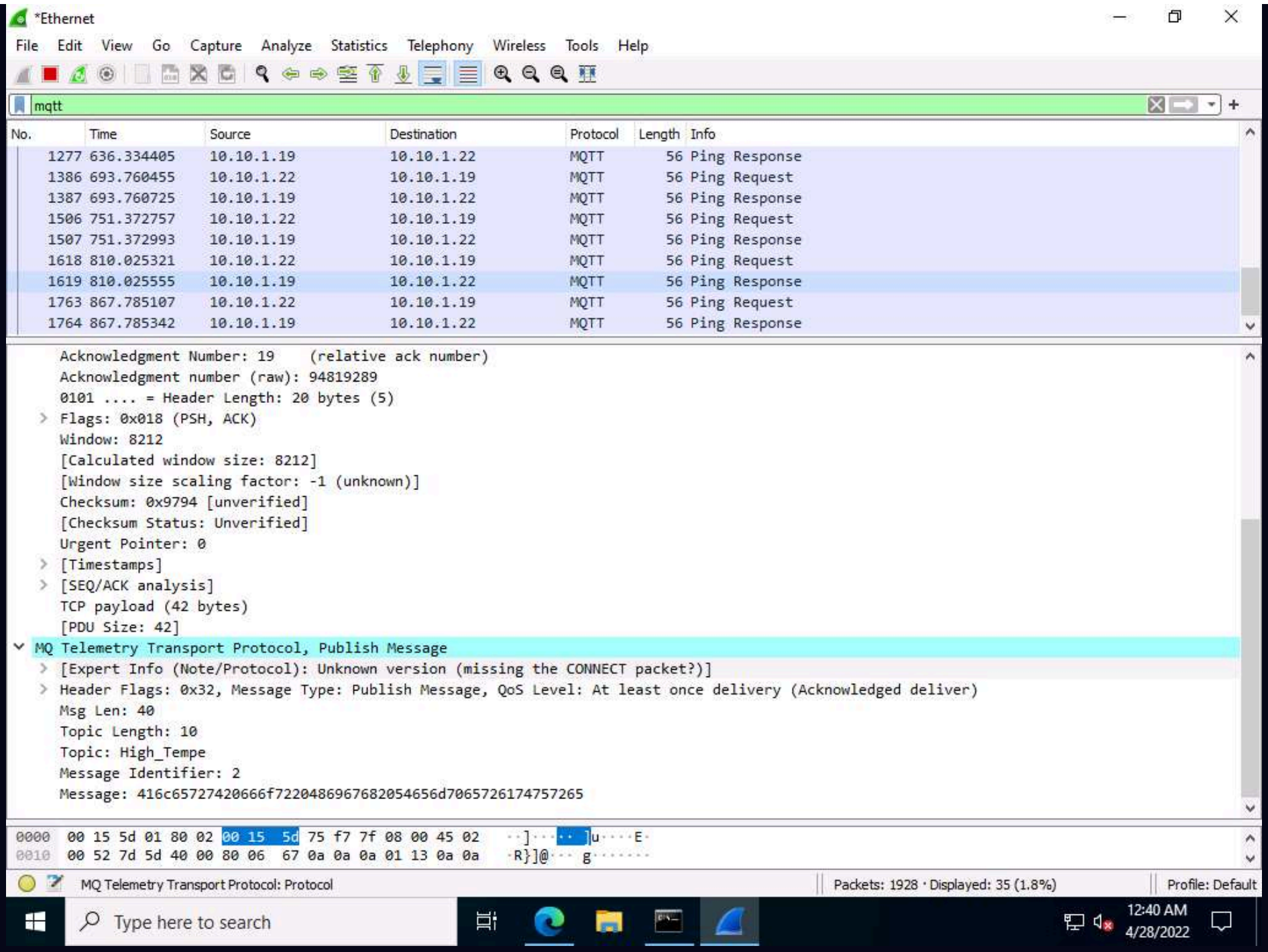
54. Type **mqtt** under the **filter** field and press **Enter**. To display only the MQTT protocol packets.



55. 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.

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

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

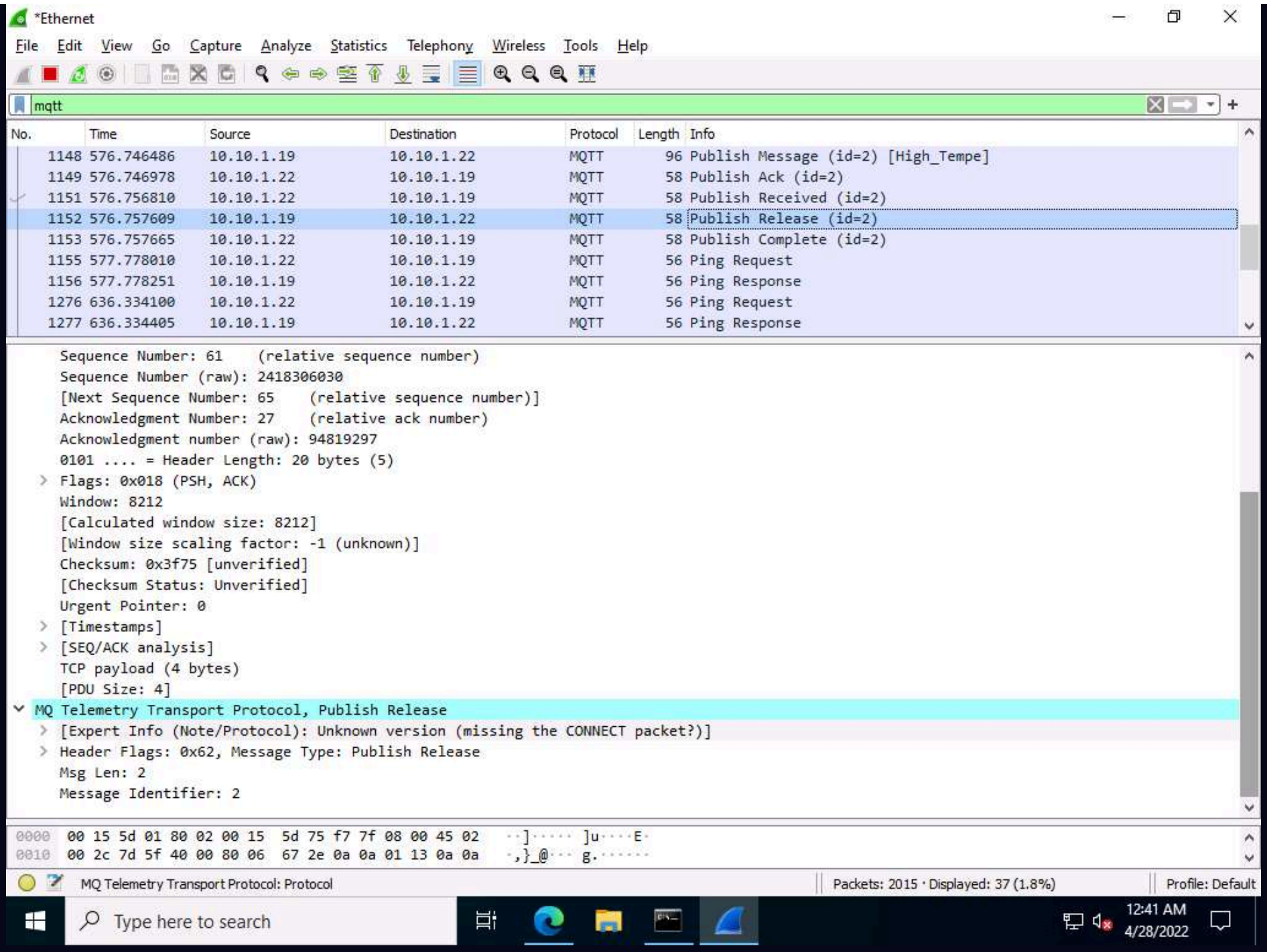


Note: Note: After establishing a successful connection with the MQTT broker, the MQTT client can publish messages. The headers in the Publish Message packet are given below:

- 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.

58. 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.

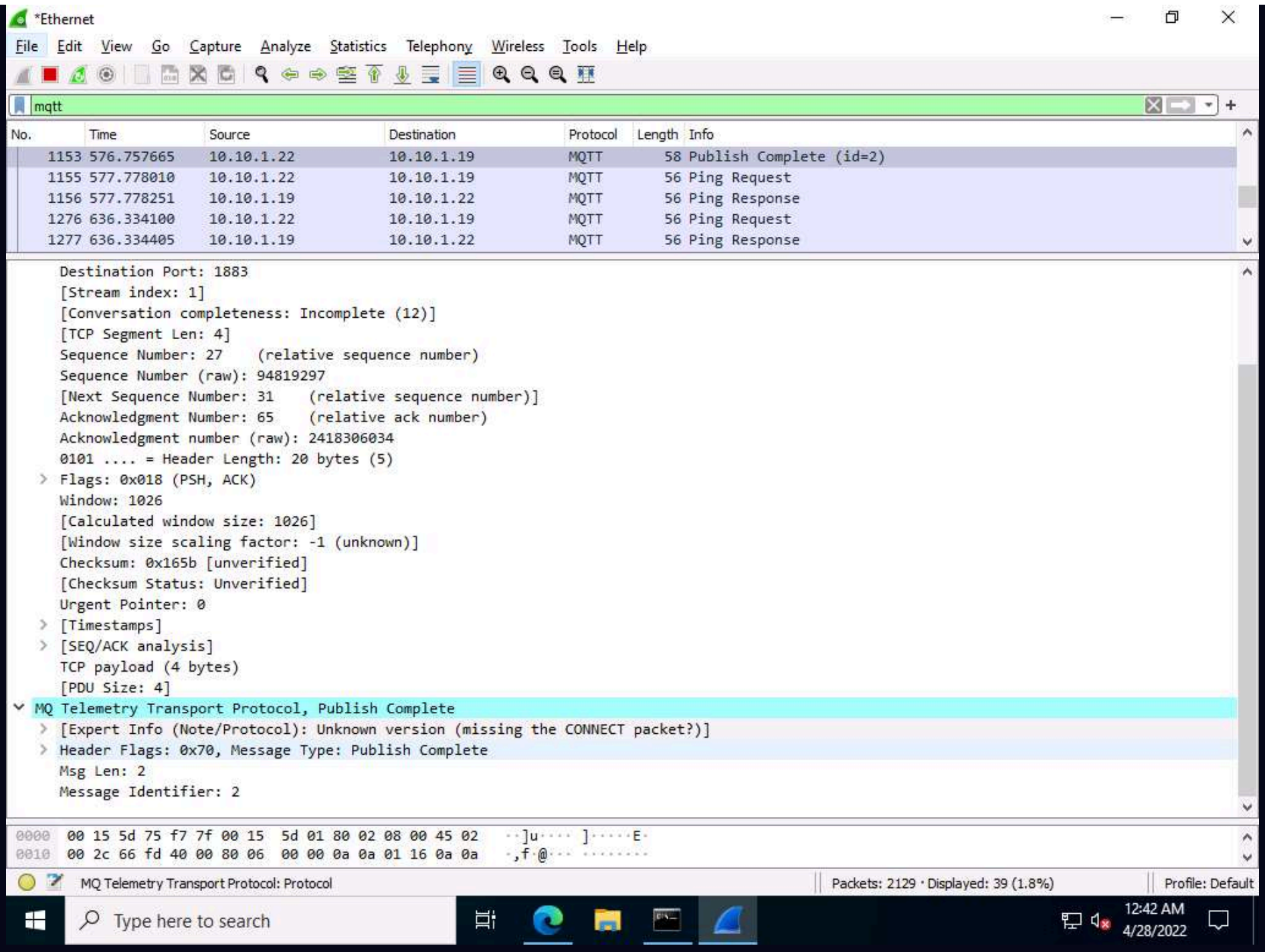
59. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Message Type**, **Message Identifier**.



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

60. 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.

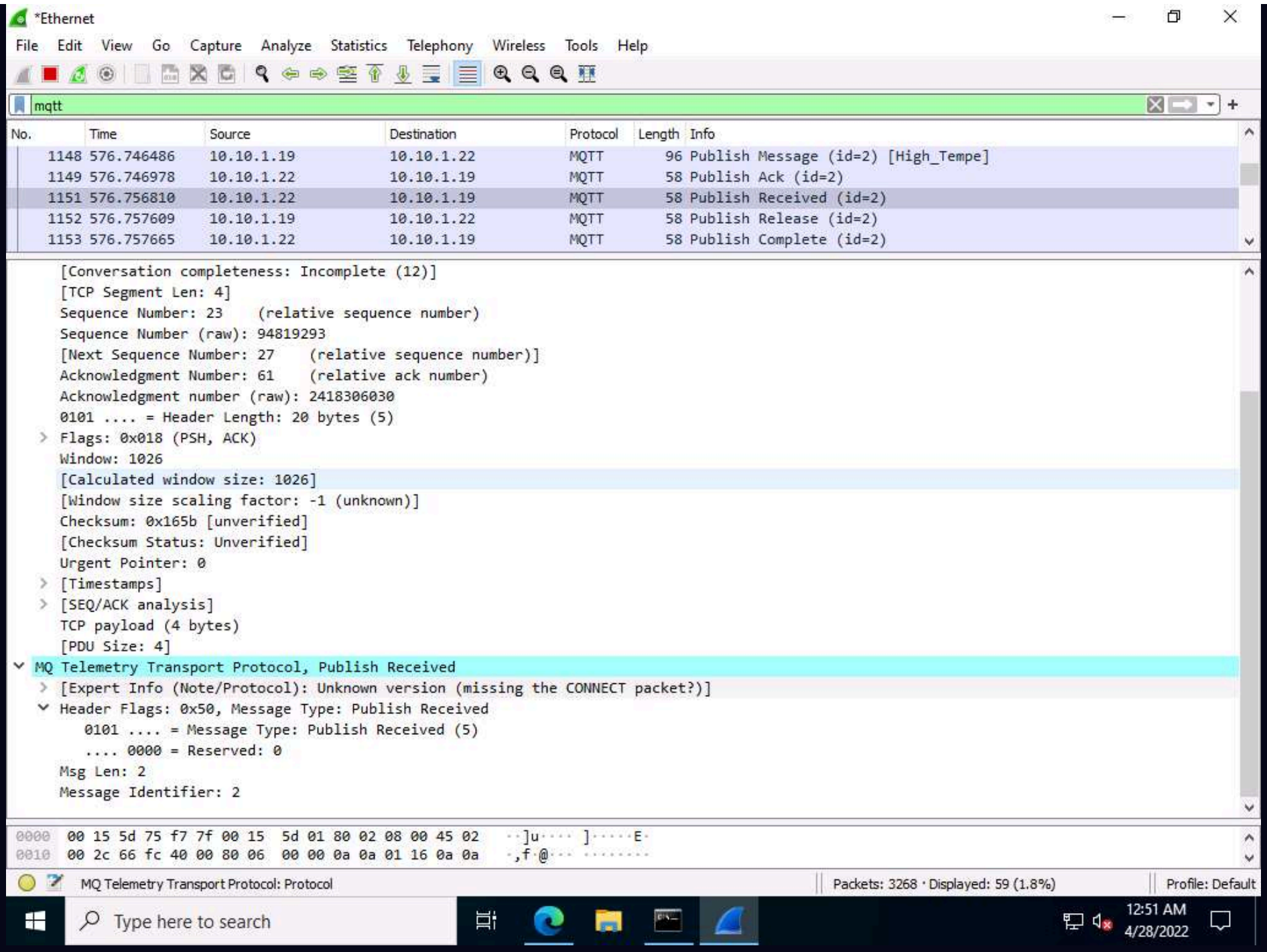
61. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len** and **Message Identifier**.



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

62. 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.

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



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

65. 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.

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