Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi

Faculty of Cyber Security 2024-2025

IOT AND OT HACKING

Performed by:

Rida Shahid Malik 2023610
Ummama Khan 2023738
Momna Jamil 2023336

Submitted To
Sir Abdullah Bin Zarshaid

Date of Submission: 14th May, 2025

OBJECTIVE

Objective

The objective of this lab is to develop practical skills in ethical hacking of Internet of Things (IoT) and Operational Technology (OT) systems. Key tasks include:

- **Performing footprinting to gather intelligence on IoT and OT devices.**
- **Capturing and analyzing network traffic between IoT devices, focusing on the MQTT protocol.**
- **Identifying vulnerabilities in IoT and OT environments through reconnaissance and simulation of attack scenarios.**

Environment Setup

Lab 1:

- **Operating System: Windows 11 Virtual Machine (VM)**

Lab 2 and Lab 3:

- **Operating Systems:**
    - **Windows 11 Home VM and 2 Host OS**
    - **Windows Server 2019 (Standard Version)**

Tools Used:

- **Wireshark: For capturing and analyzing network traffic.**
- **Shodan: For discovering internet-connected IoT devices and identifying open ports.**
- **Bevywise MQTTRoute: For setting up and managing an MQTT broker.**

Lab Tasks

Lab 1: Footprinting and Reconnaissance

1. **Step 1: Virtual Machine Setup:**
    - **Configured a Windows 11 VM to serve as the primary environment for reconnaissance tasks.**

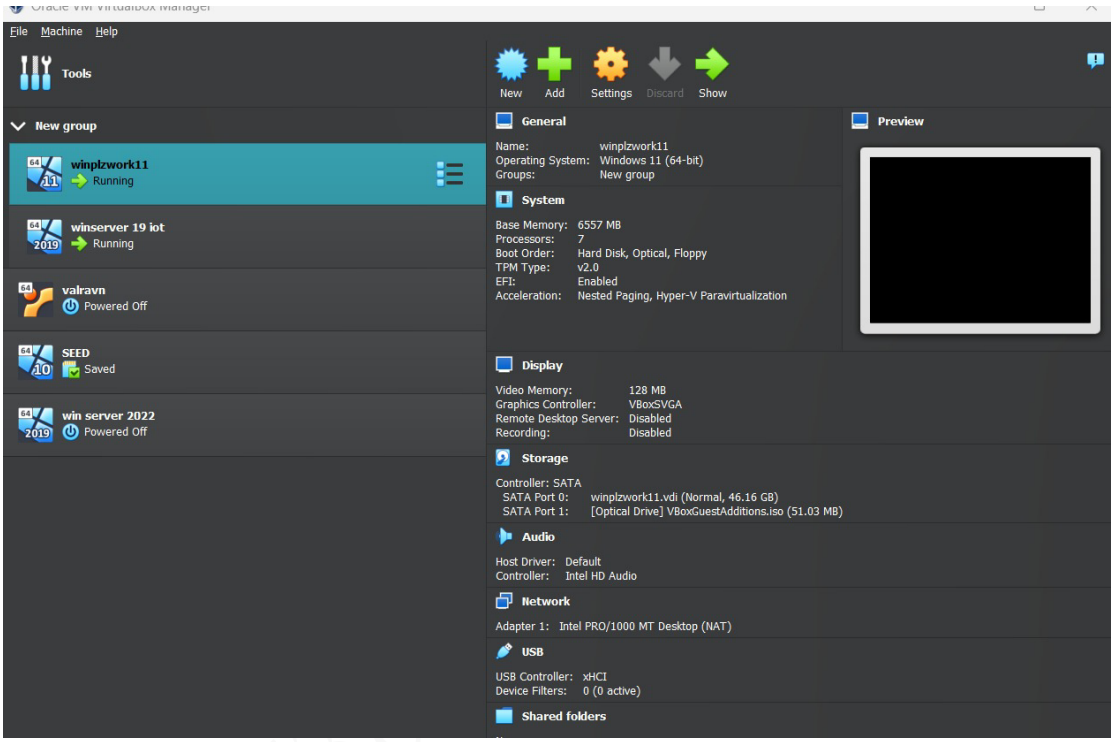


Figure 1 VM configuration for windows 11 home

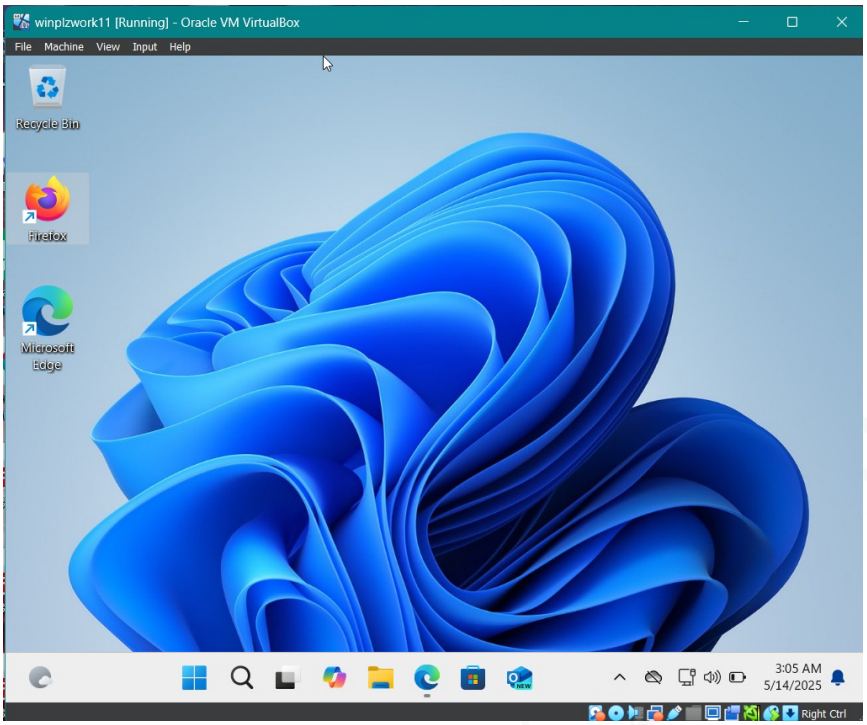


Figure 2 what a successfully running VM should look like

Step 2: Whois Lookup:

- Accessed whois.com/whois/ via a web browser.

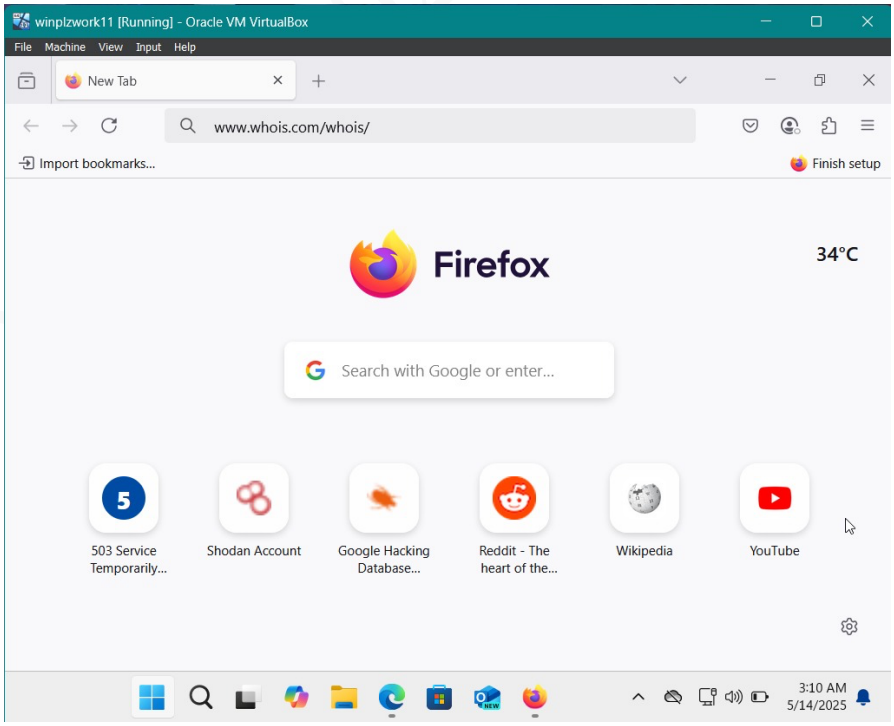


Figure 3 access whois.com using any web browser

- Searched for oasis-open.org to gather domain-related information, including registrar details, IP addresses, and contact information.

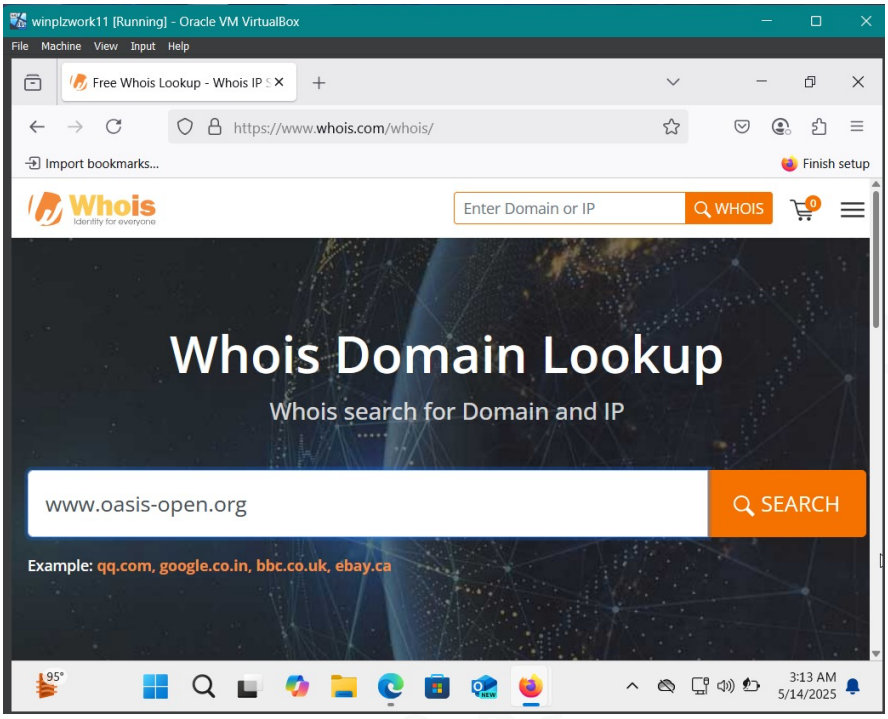


Figure 4 on whois.com search for oasis-open.org

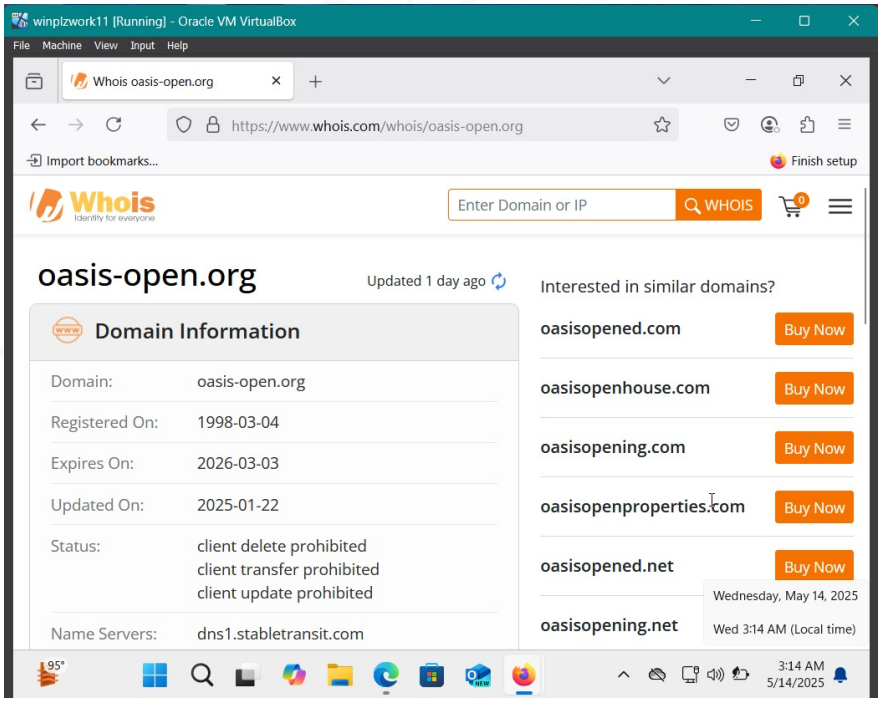


Figure 5 gather information on the domain you looked up

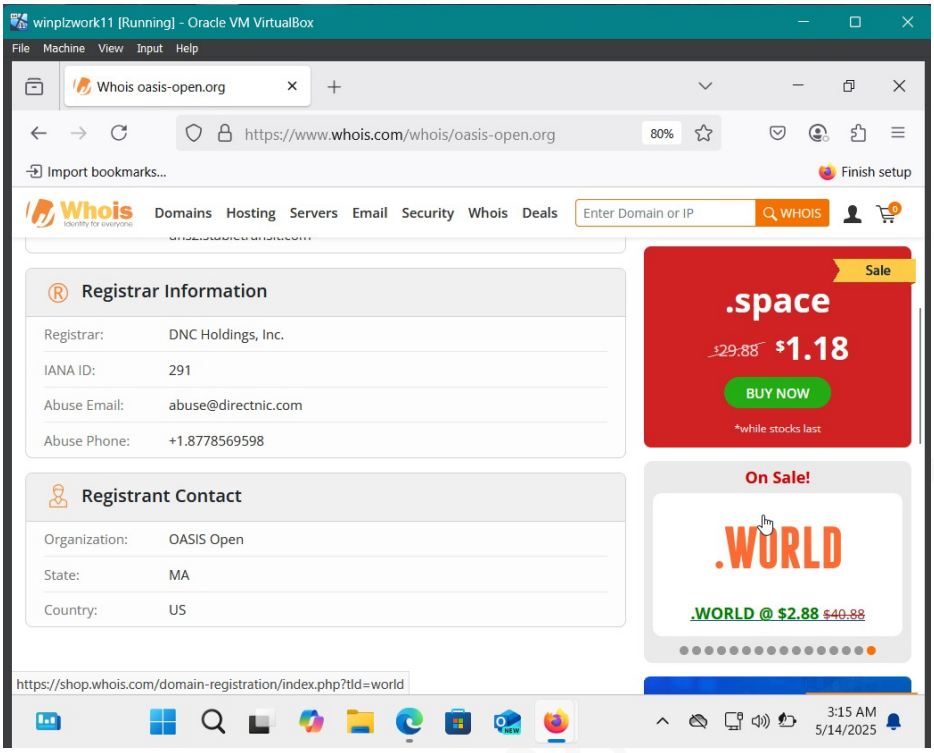


Figure 6 oasis-open.org further information

Step 3: Open www.exploit-db.com/google-hacking-database:

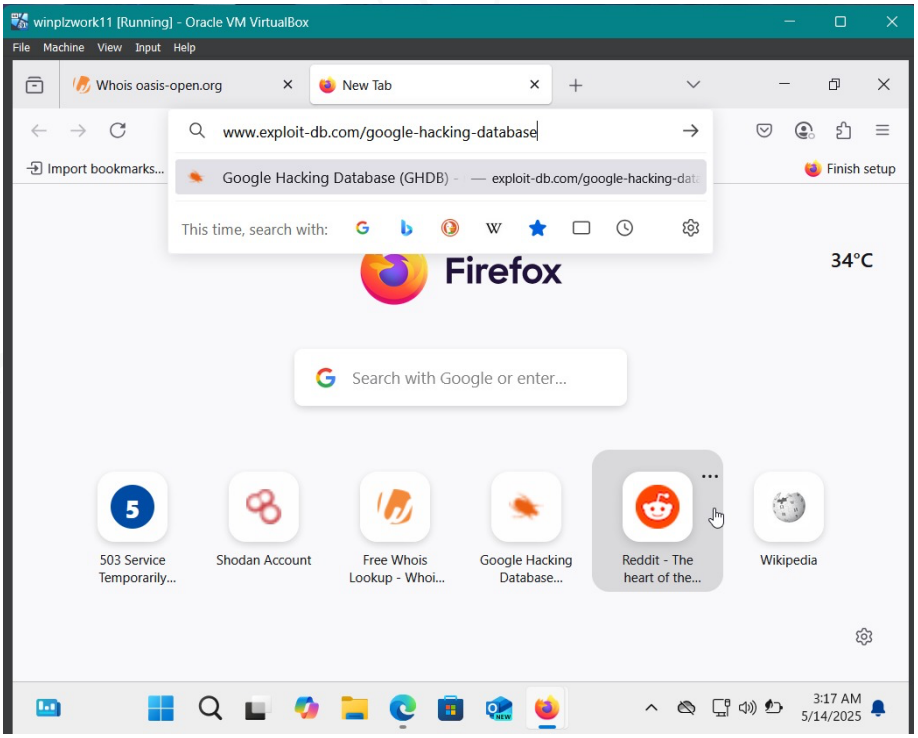


Figure 7 using your web browser access the google exploit database

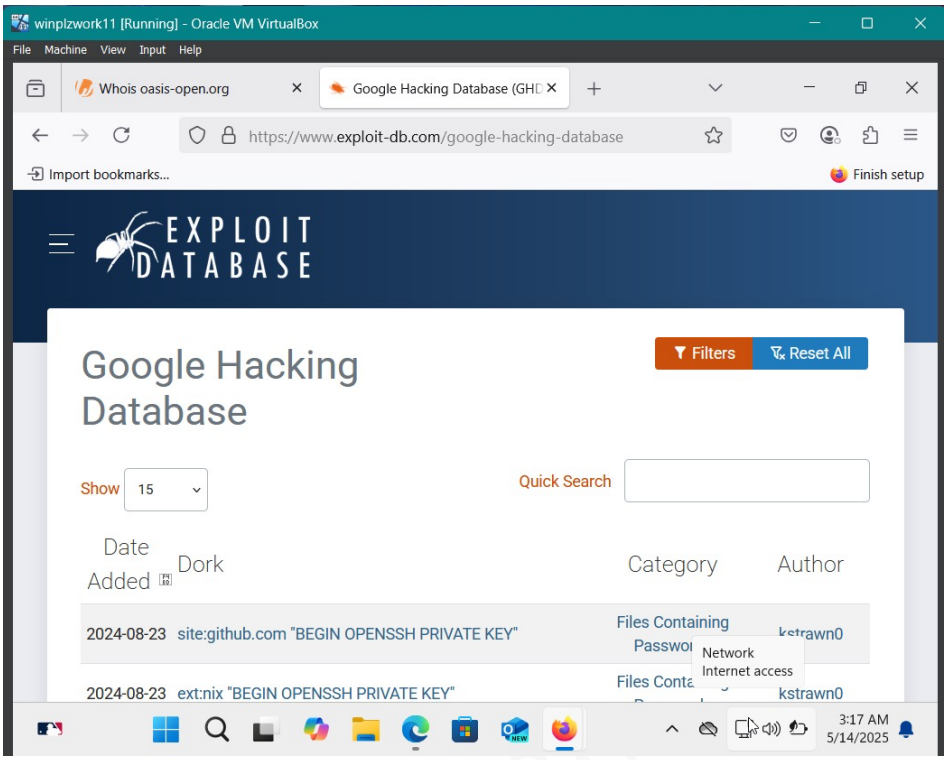


Figure 8 google exploit database interface

Step 4: Enter scada in the Quick Search:

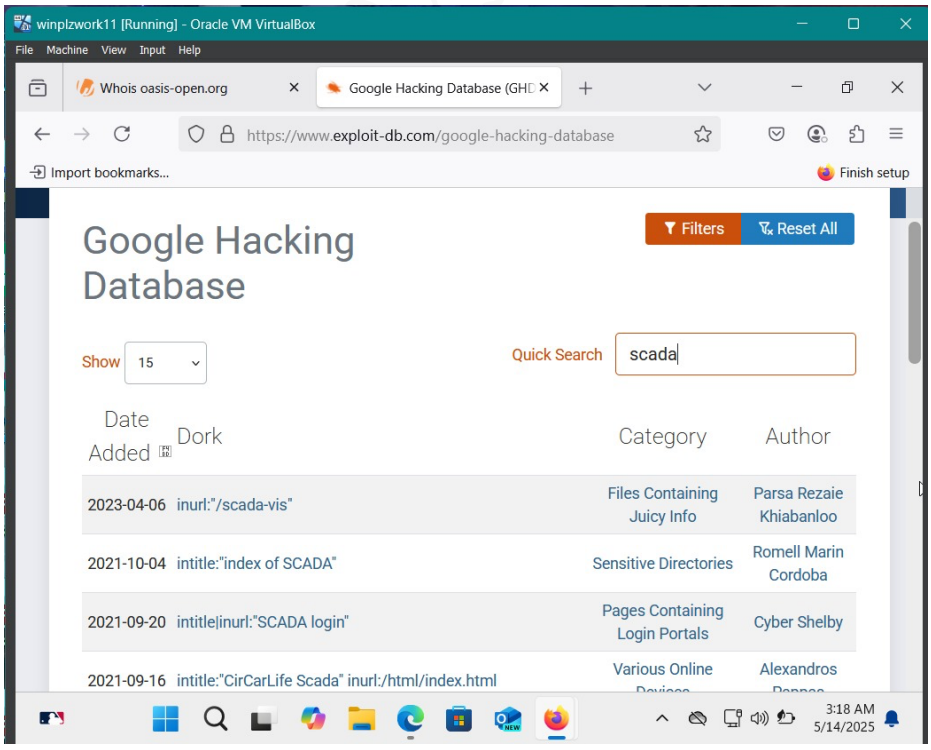


Figure 9 search for scada in the exploit database to find various scada 'dorks'

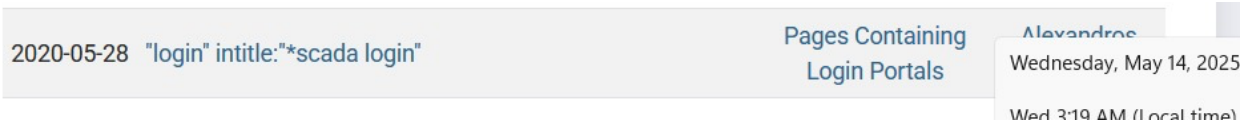


Figure 10 pick any given 'dork' to start exploiting

Step 5: Copy this to Google:

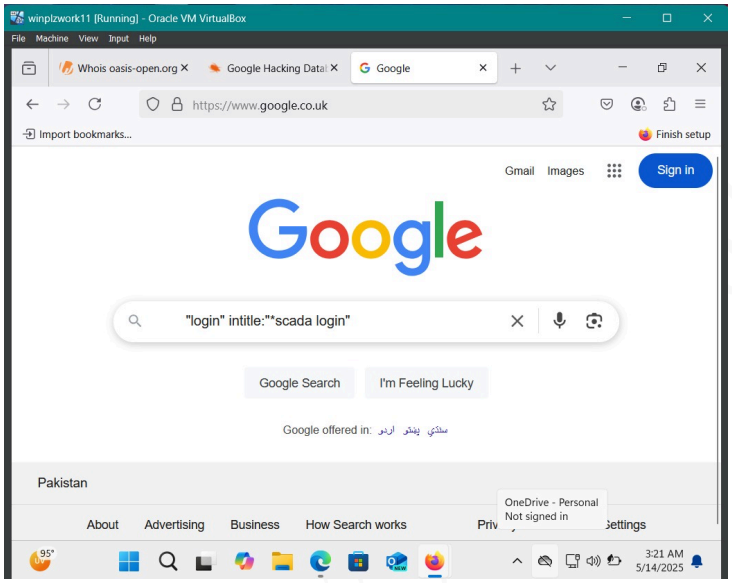


Figure 11 paste the google dork onto the google search engine

Step 6: Google Dork Search Result for SCADA Login Portal:

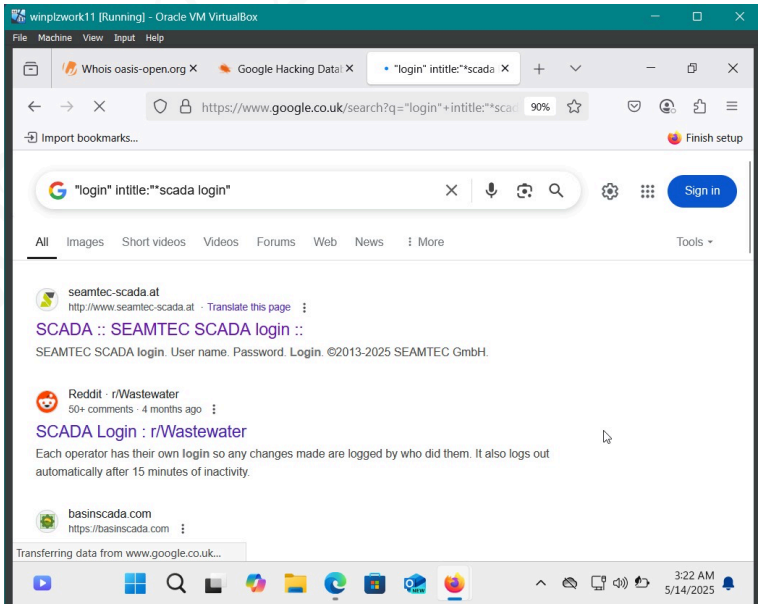


Figure 12 google dork search results

Step 7: Select the Seamtec Scada Login (as specified in Lab Manual)

Figure 13 find any particular vulnerable login information

- The Scada Login was vulnerable to brute-force attacks due to weak authentication mechanisms.

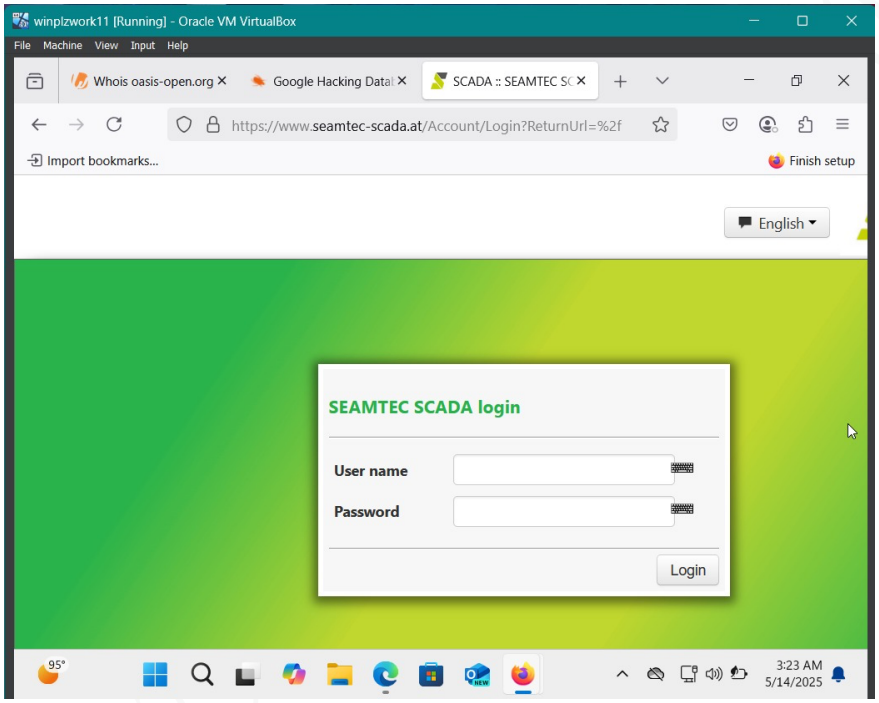


Figure 14 vulnerable login interface

Step 8: Open Shodan:

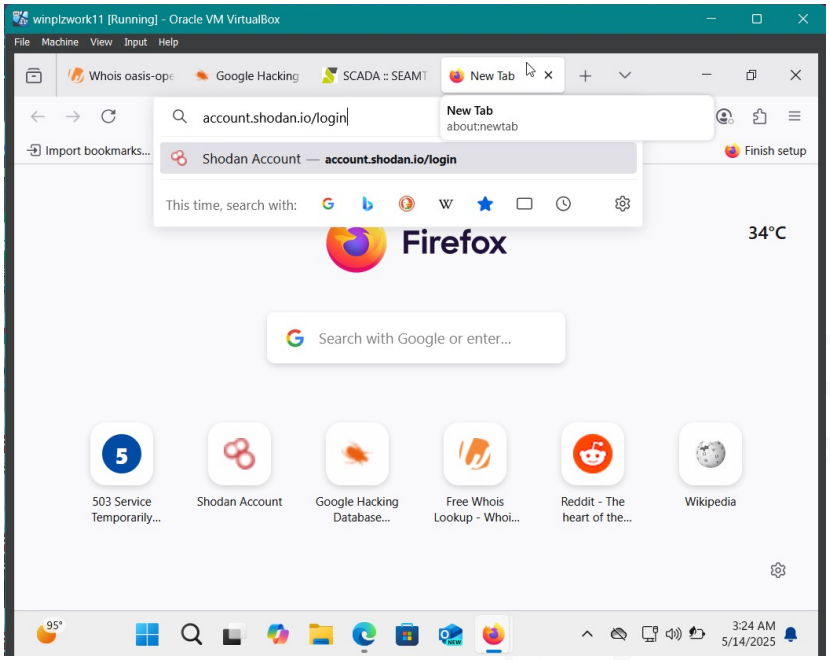


Figure 15 using a web browser access shodan

Step 9: Create Account:

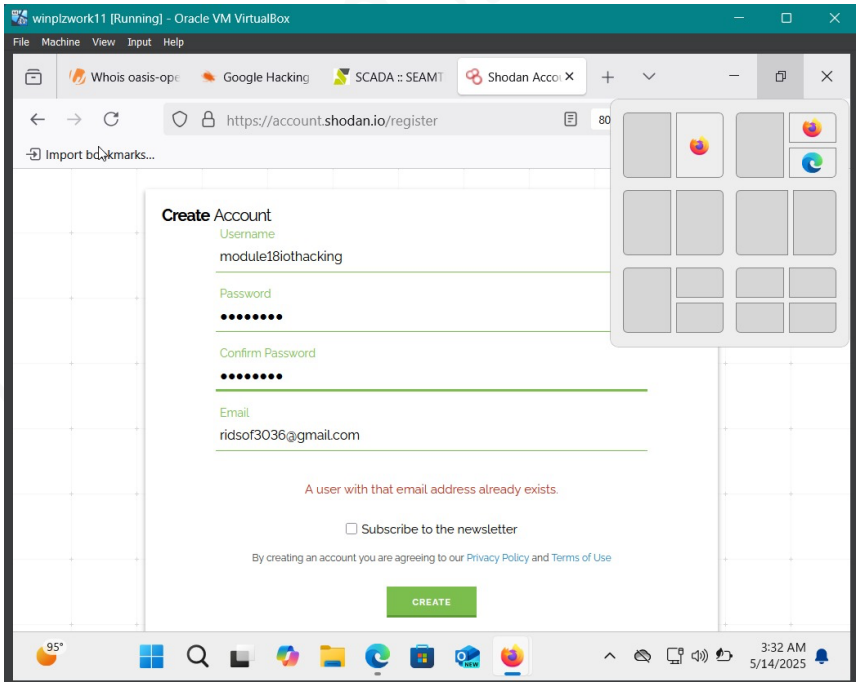


Figure 16 login to shodan by creating an account

Step 10: Login:

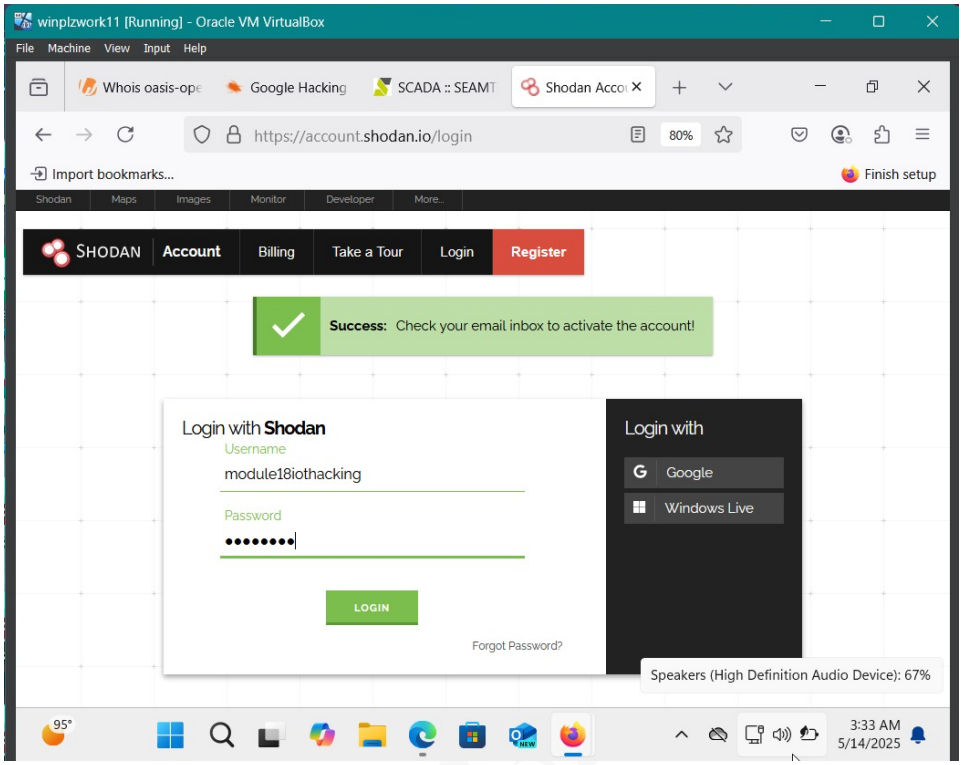


Figure 17 use account creation credentials to finally login

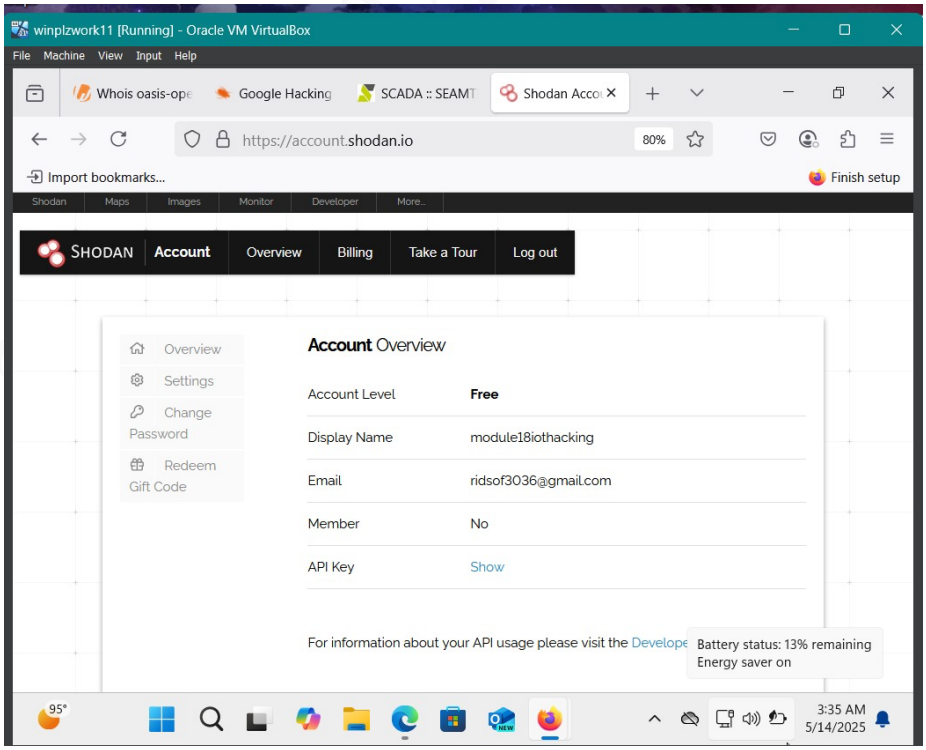


Figure 18 shodan interface

Step 11: Open Dashboard and search for devices using the query port:1883 to identify MQTT-enabled IoT devices.

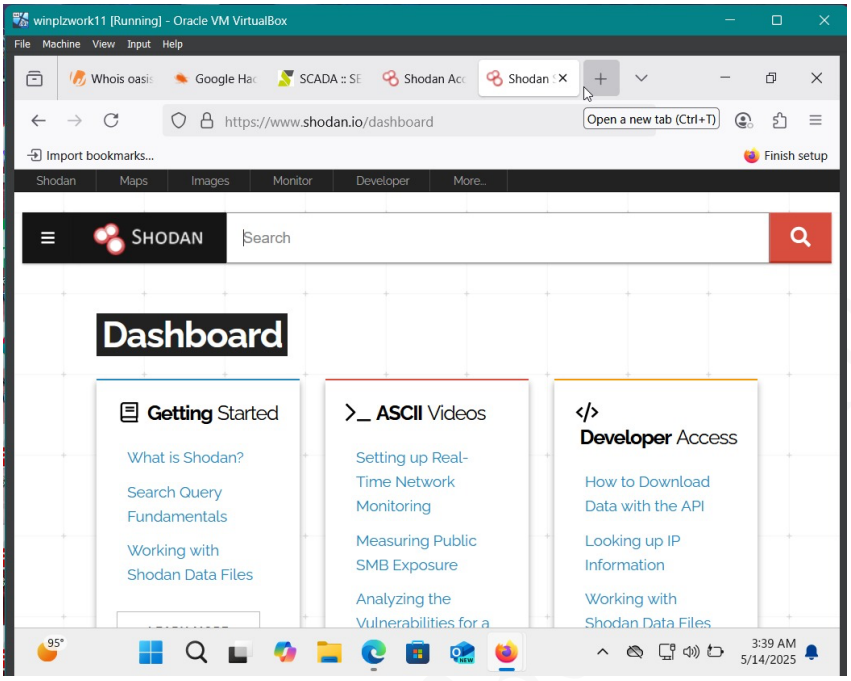


Figure 19 to exploit iot vulnerabilities we'll search for the mqtt enabled iot devices on the shodan dashboard using the query for port 1883

Step 12: Analyzed the Shodan dashboard, which displayed open ports, device locations, hostnames, organizations, and supported protocols.

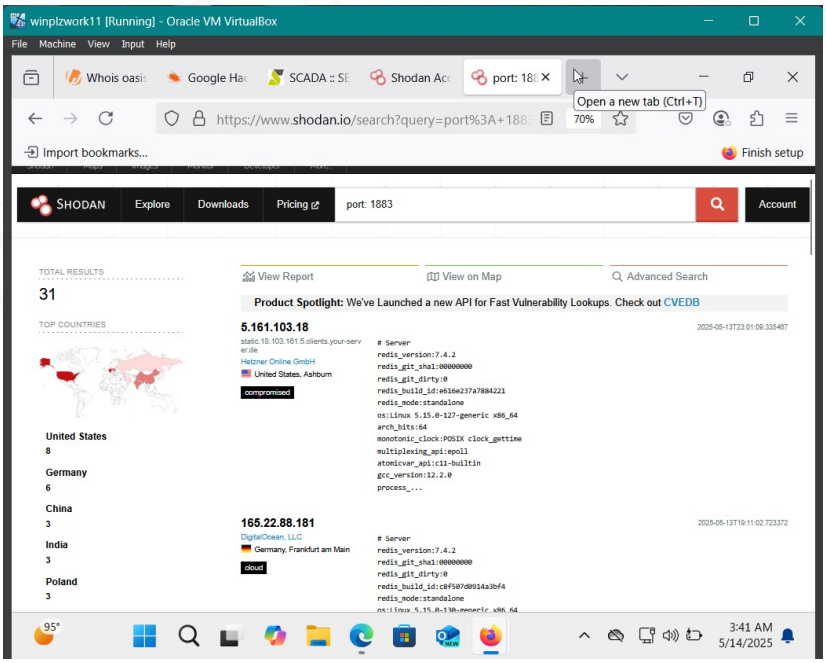


Figure 20 port 1883 search query results

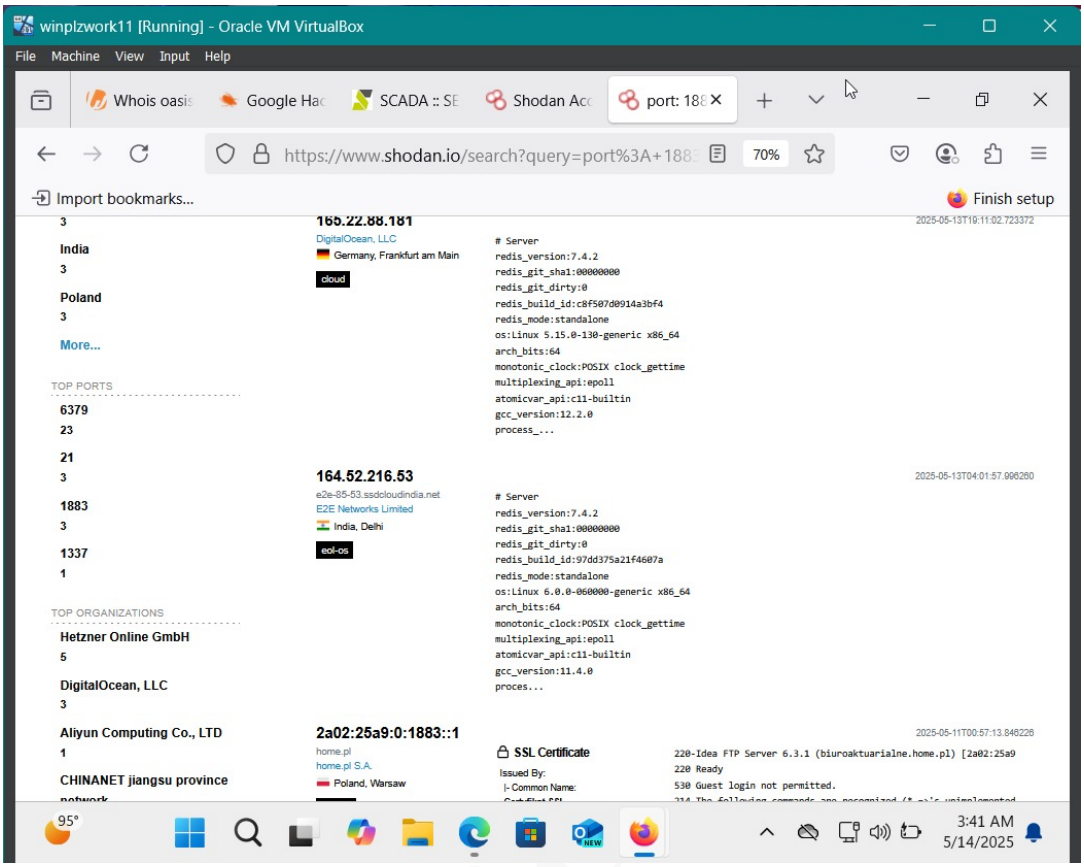


Figure 21 port 1883 search results continued

Step 13: Selected a target IP for an MQTT device and reviewed its detailed report, including geolocation and service banners.

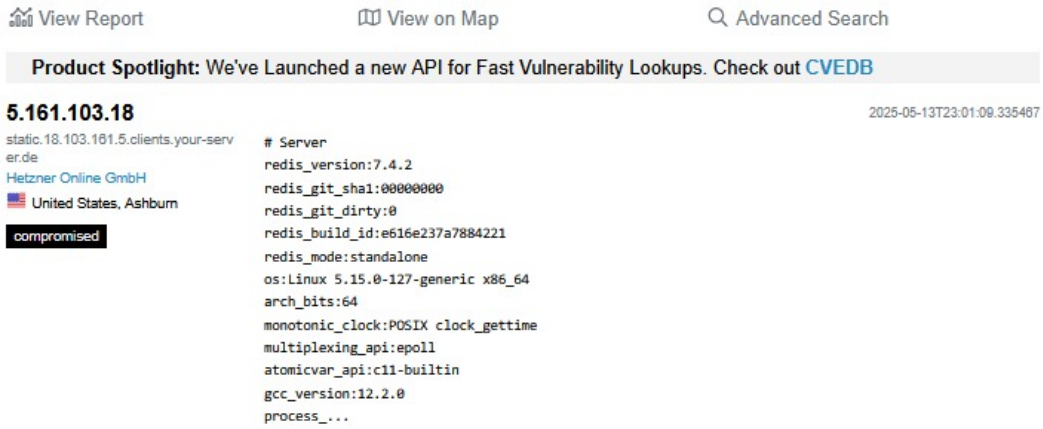


Figure 22 select any target ip and review its details

All information displayed openly (all open ports, location, host name, organization, protocols, etc.)

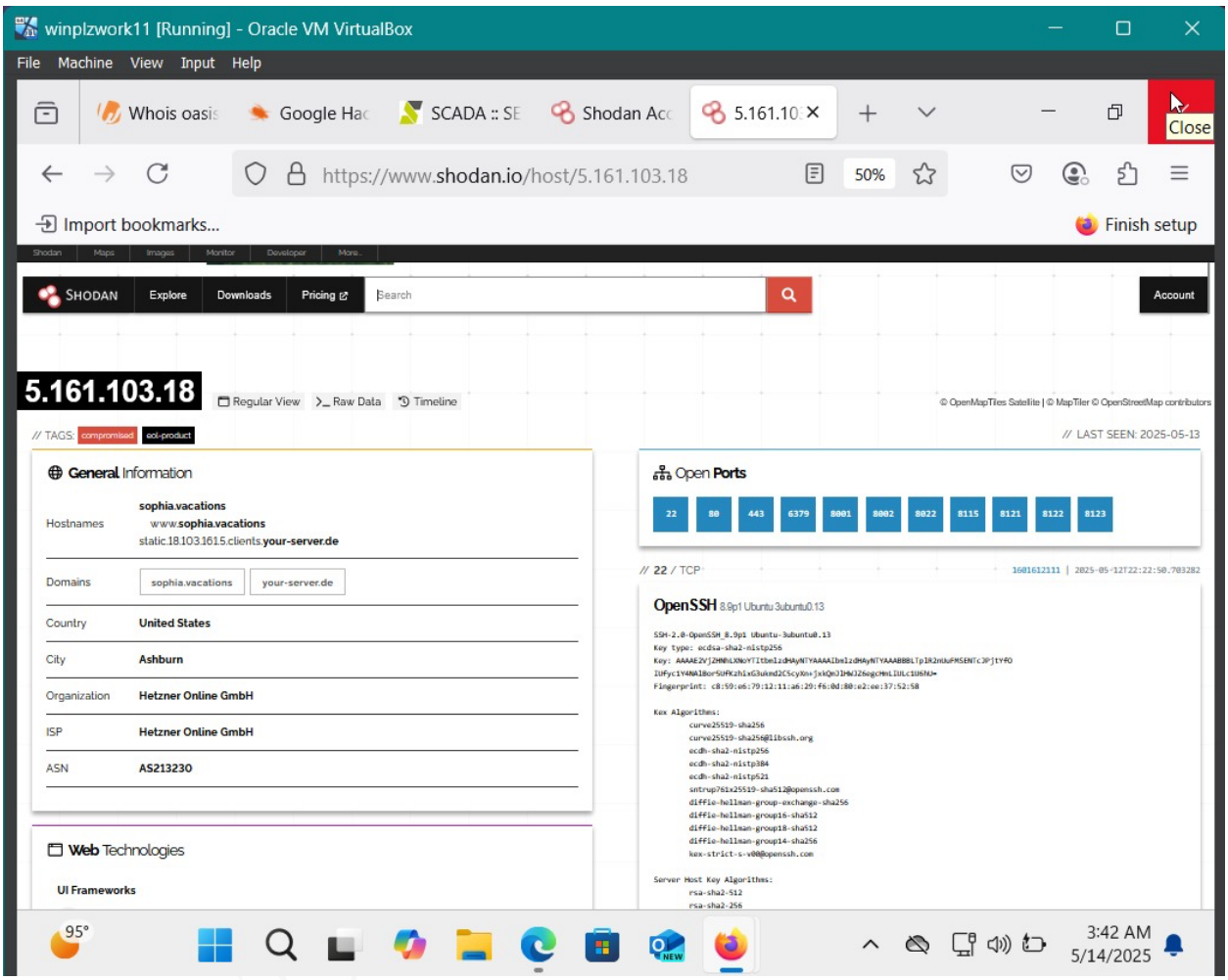


Figure 23 ip information displayed on shodan

Lab 2

Step 1: click on the mqttRoute app (your broker for this module) installed which will open the following in the command prompt:

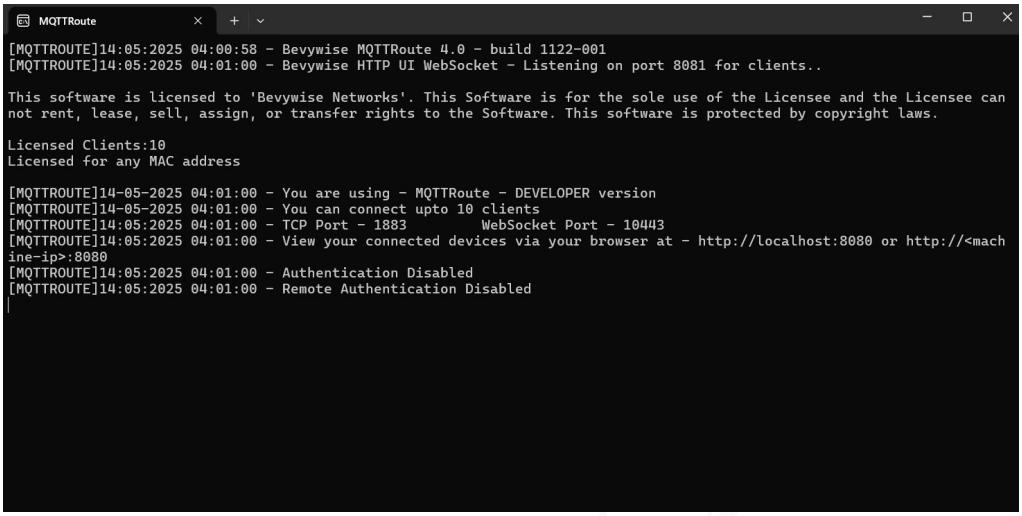


Figure 24 cmd mqttroute display

Step 2: Open command prompt (win + R)

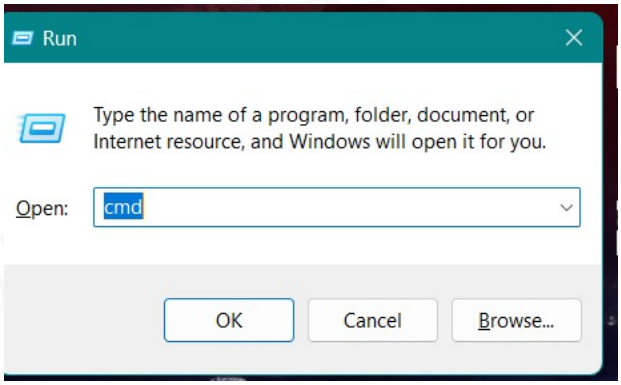


Figure 25 to access the command prompt click WIN + R and type cmd before hitting enter

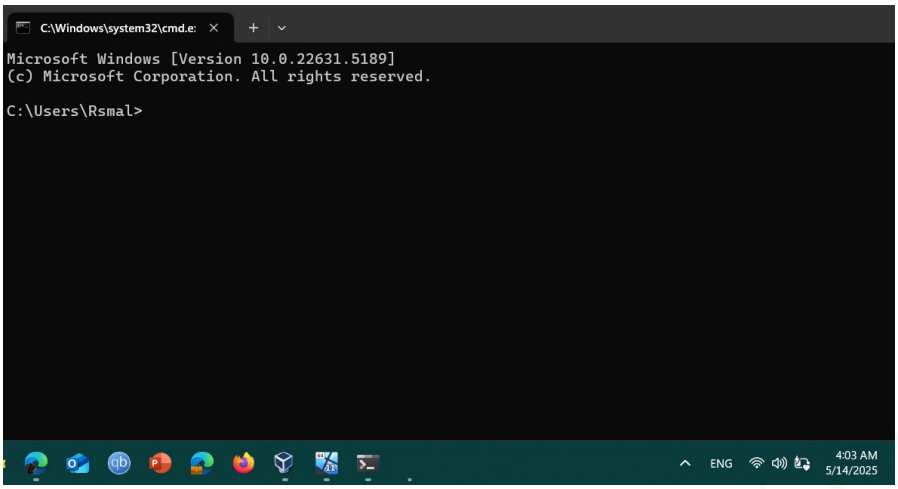


Figure 26 cmd interface

Step 3: in your command prompt type the following command and then use the ethernet adapter ipv4 address

```
C:\Users\Rsmal>ipconfig /all
```

The IP to be used:

```
Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter
   Physical Address. . . . . . . . . : 0A-00-27-00-00-0C
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.56.1(Preferred)
```

Figure 27 we used the ip 192.168.56.1 as our broker ip

Step 4: Open Web UI

Figure 28 to access the mqttRoute web UI type your machine ip followed by the port 8080 ' <Host ip>:8080'

Accessed the MQTTRoute web interface using the broker's IP address.

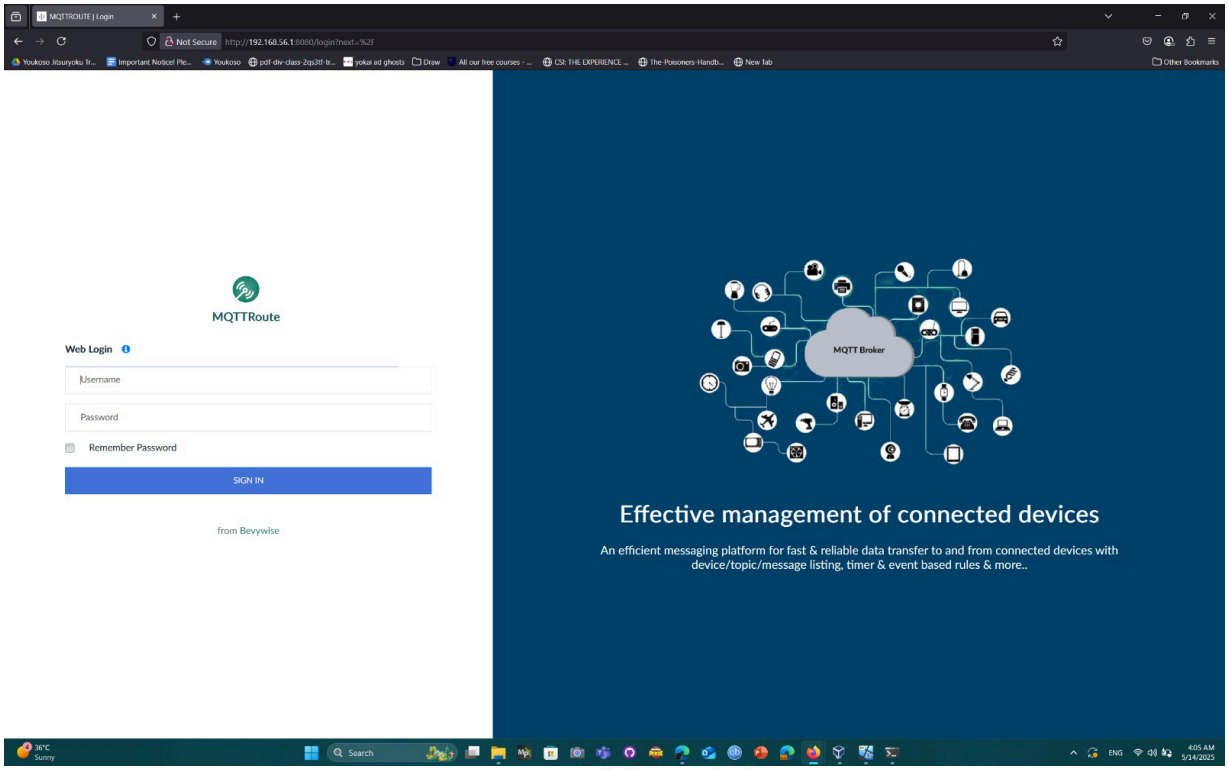
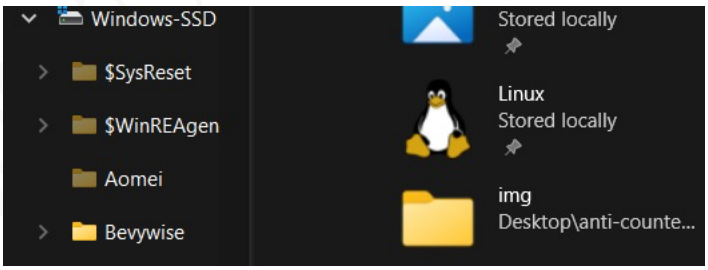


Figure 29 the broker interface fully loaded

Step 5:Log in with the default admin credentials (username: admin, password retrieved from the broker configuration file).

Step 5.1: locate the bevywise folder and enter the config folder



Step 5.2: open the config file in your notepad

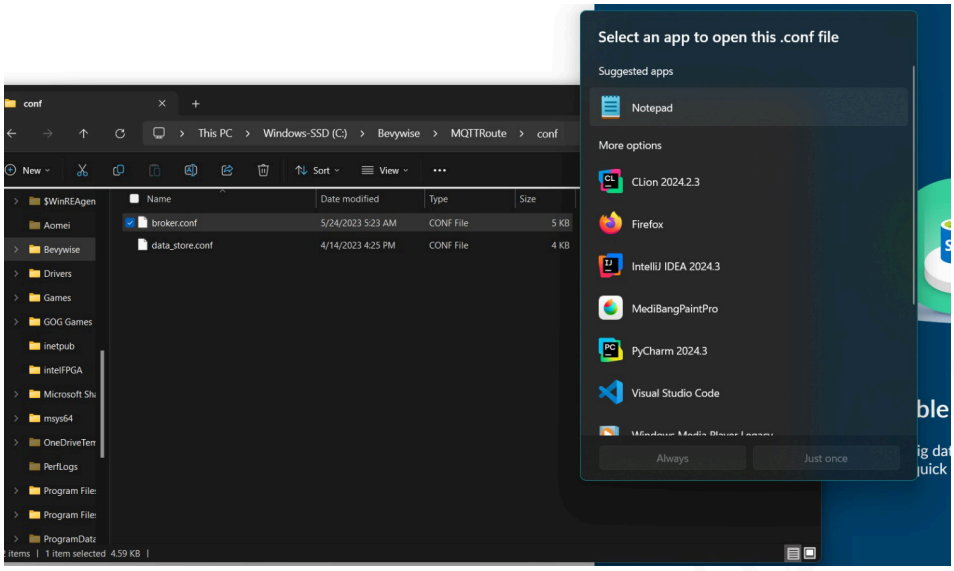


Figure 30 open the broker config file located in the bevywise config folder in your notepad to read

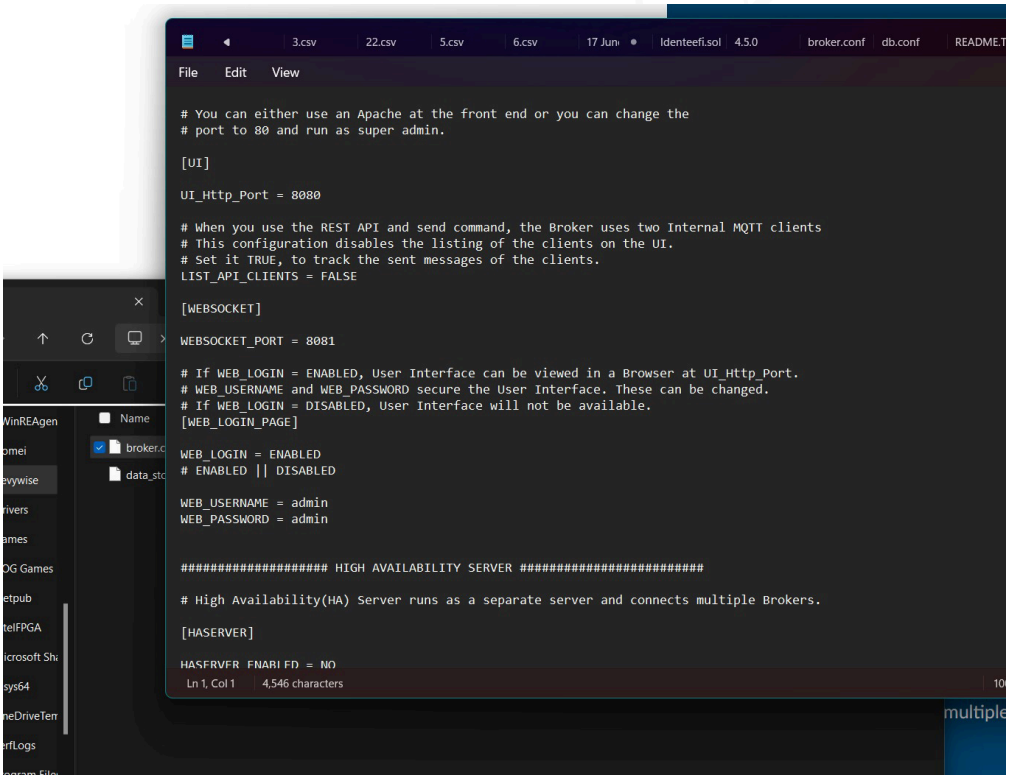


Figure 31 broker config file information including default admin credentials

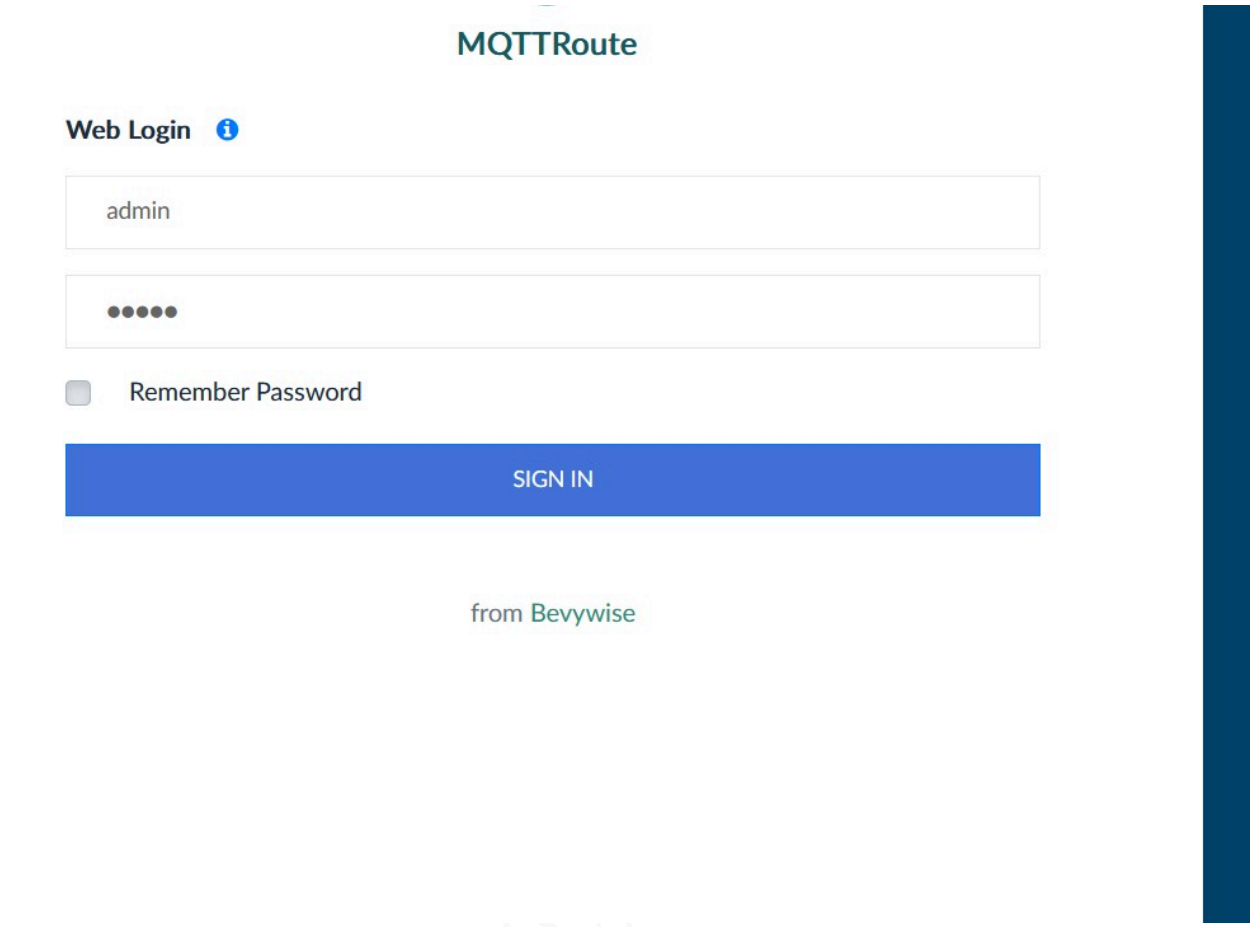


Figure 32 enter the admin credentials aka username: admin , password: admin in the mqttroute login page

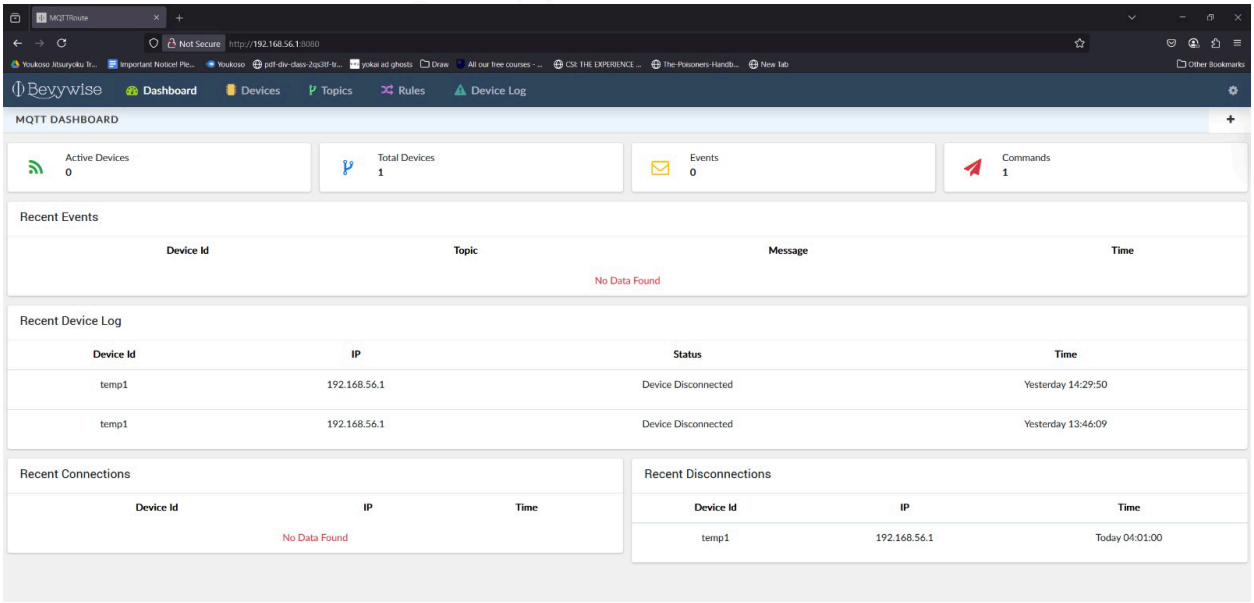


Figure 33 mqttRoute dashboard

Step 6: open the window 2019 server vm running windows 10

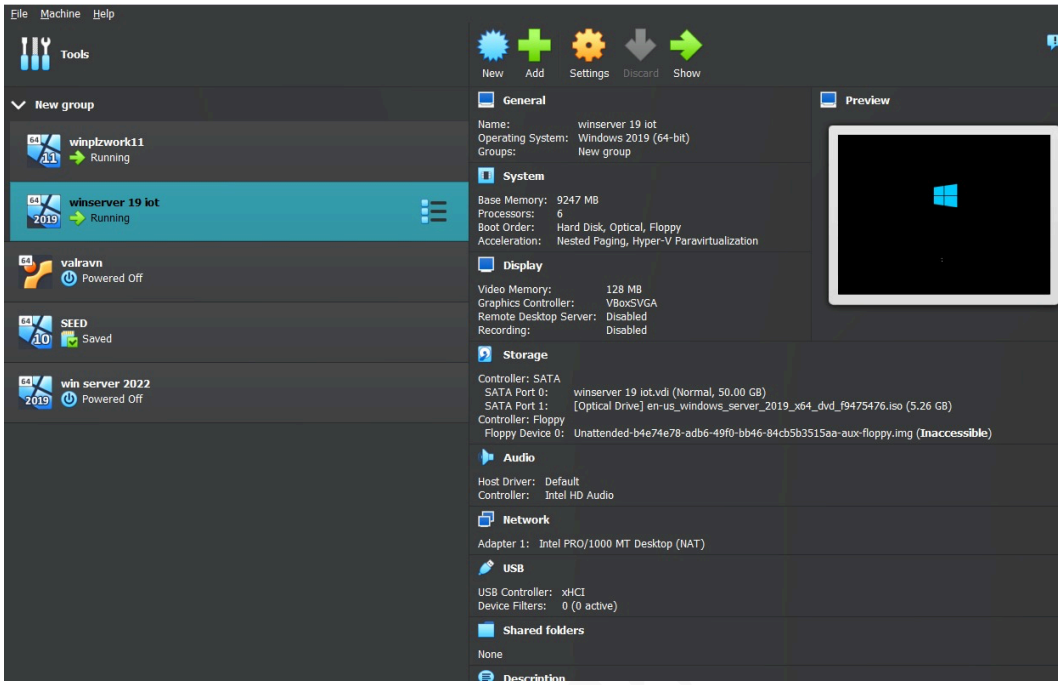


Figure 34 windows 2019 server vm (standard edition)

Figure 35 to unlock select insert , keyboard followed by the Insert CTRL+ALT+DEL option

Step7: download the bevywise iot simulator and then enter the bevywise files locating the bin folder before running the runsimulator.bat file

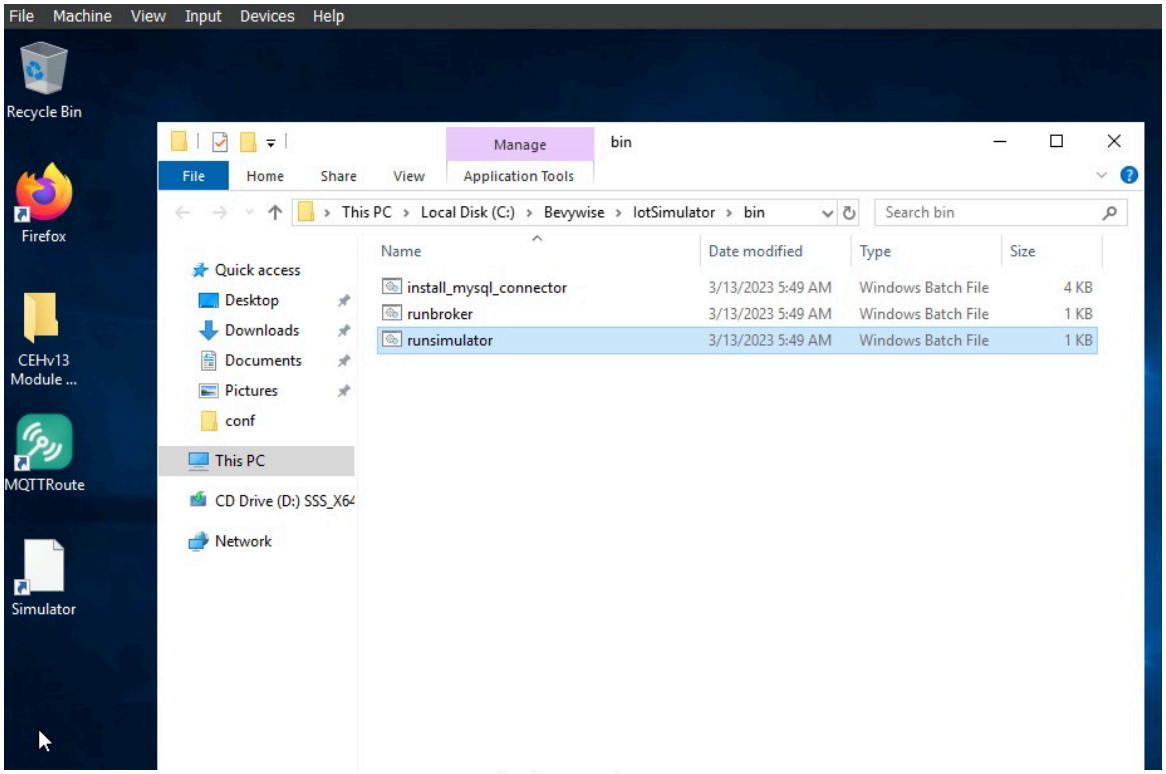


Figure 36 locate the runsimulator.bat file

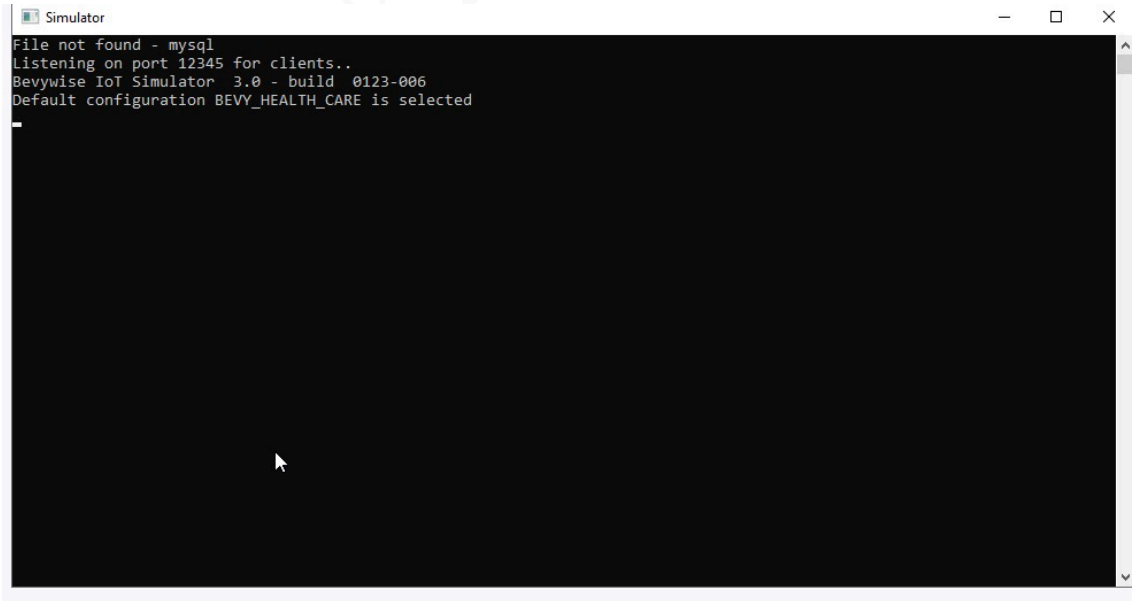


Figure 37 terminal that opens upon running the runsimulator.bat

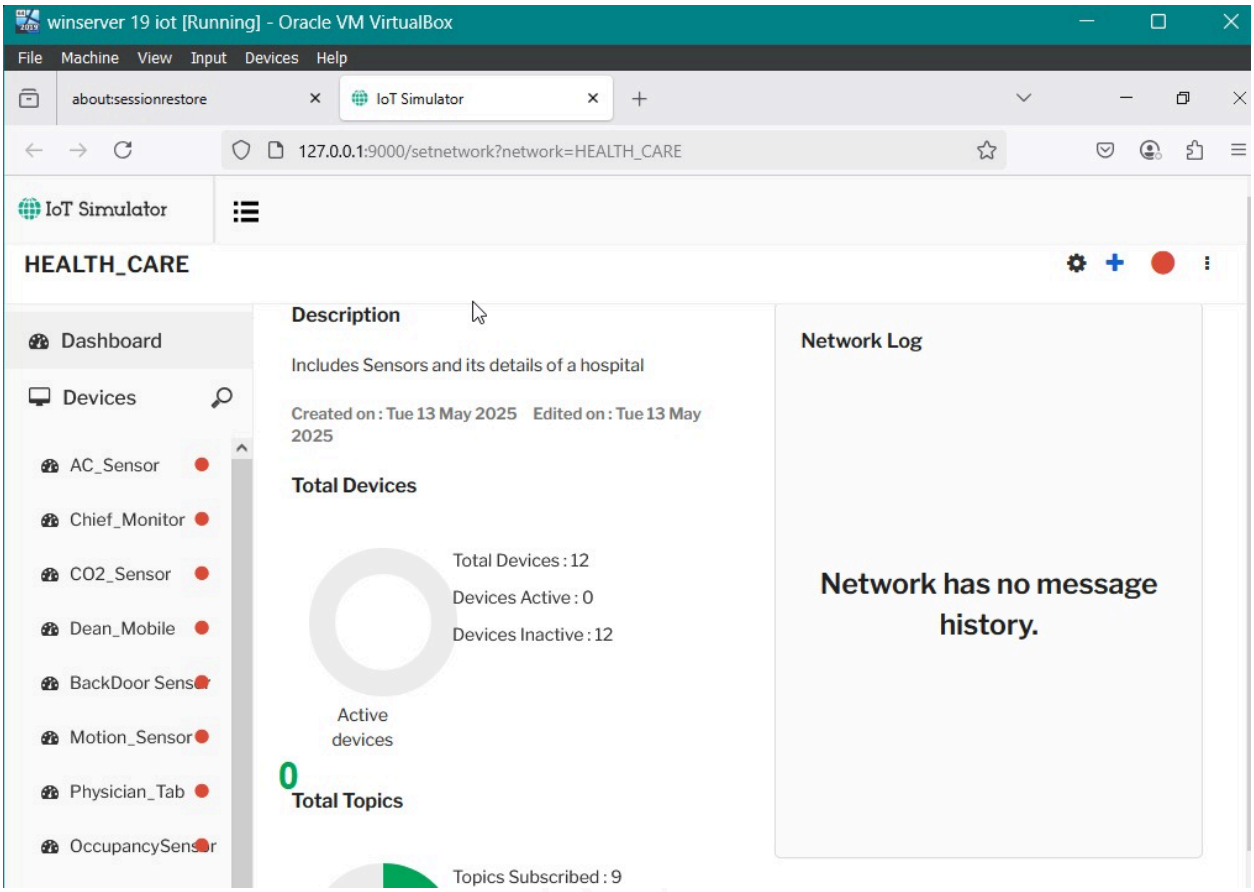


Figure 38 bevywise iot simulator ui interface

Step 8: create a new network

step 8.1: hit the create network option

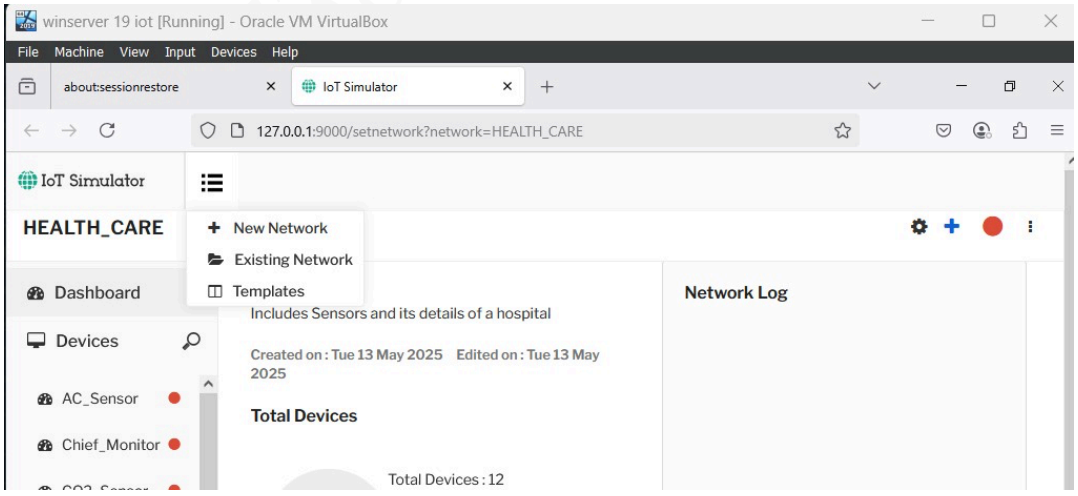


Figure 39 create a new network by hitting the menu button on the left followed by new network

Step 8.2: name the network

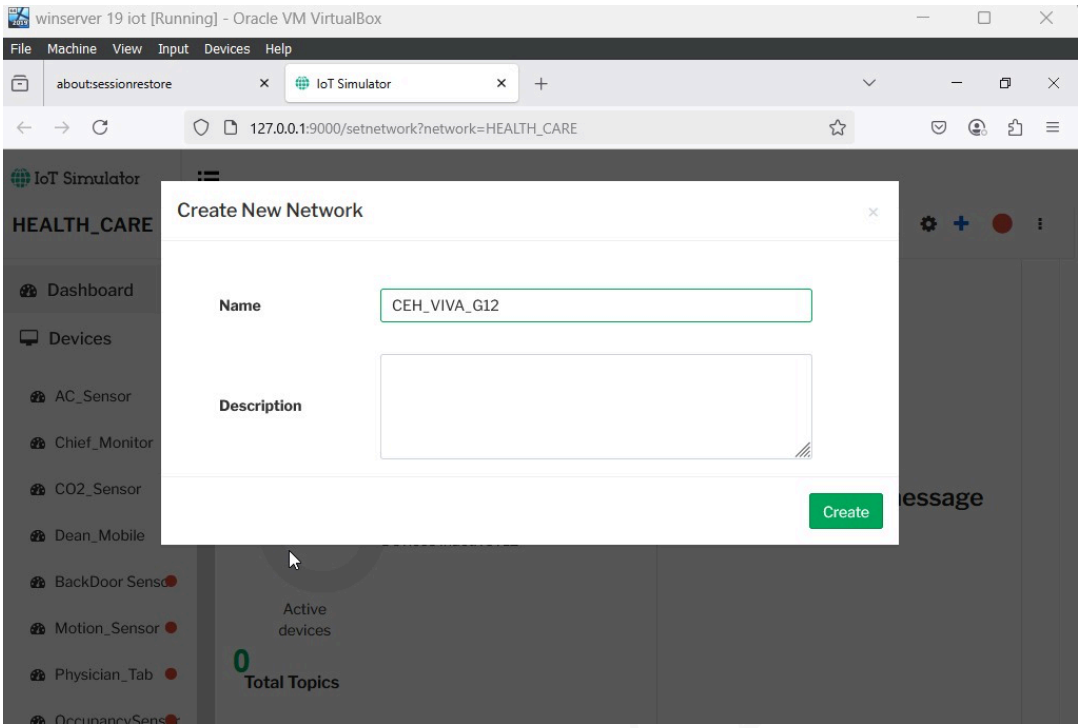


Figure 40 name the network (no spaces)

Step 8.3: replace broker ip

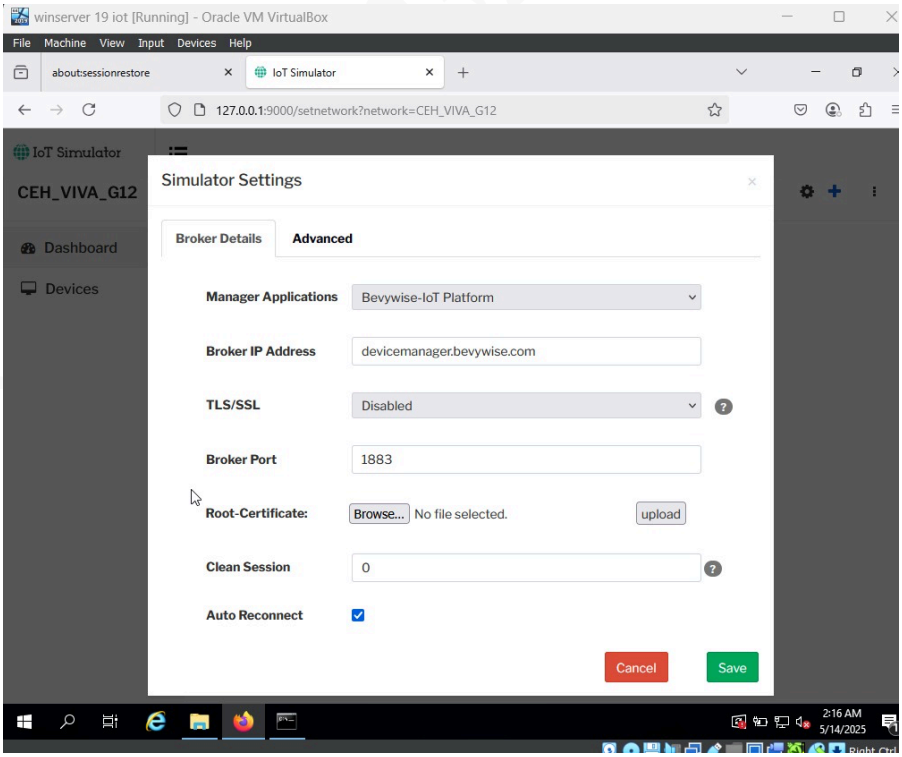


Figure 41 once you name the network replace the broker ip address with the ip used earlier

Added the broker IP to the MQTT client configuration, ensuring the port matched the terminal CLI output (default: 1883).

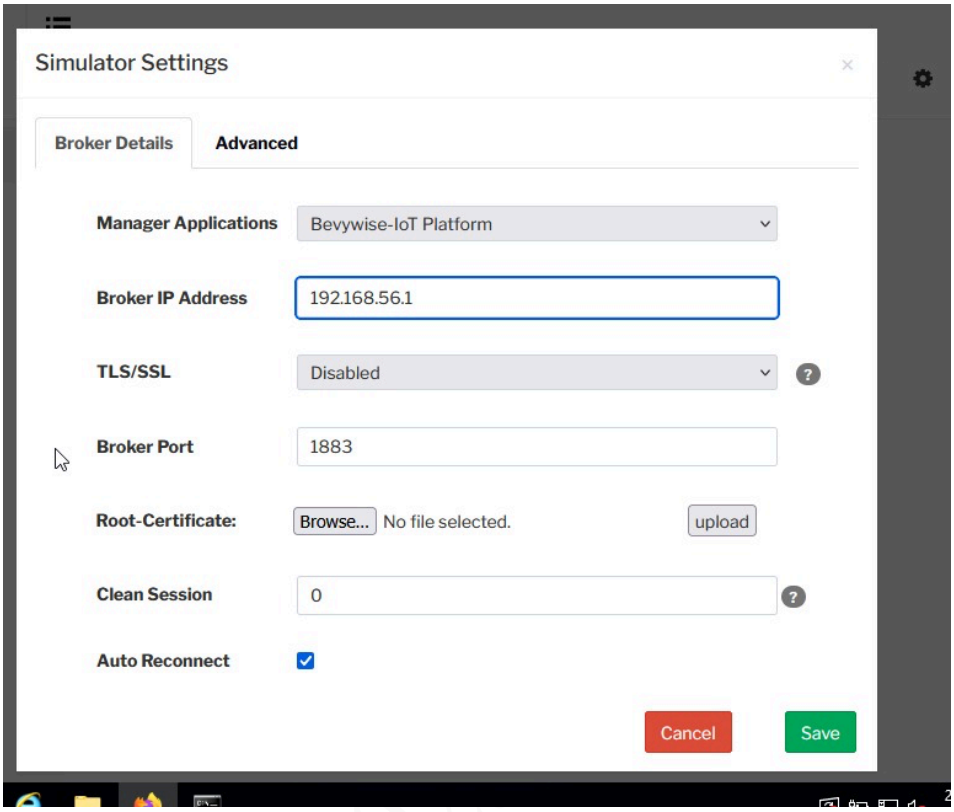


Figure 42 broker ip replaced with the ip used prior

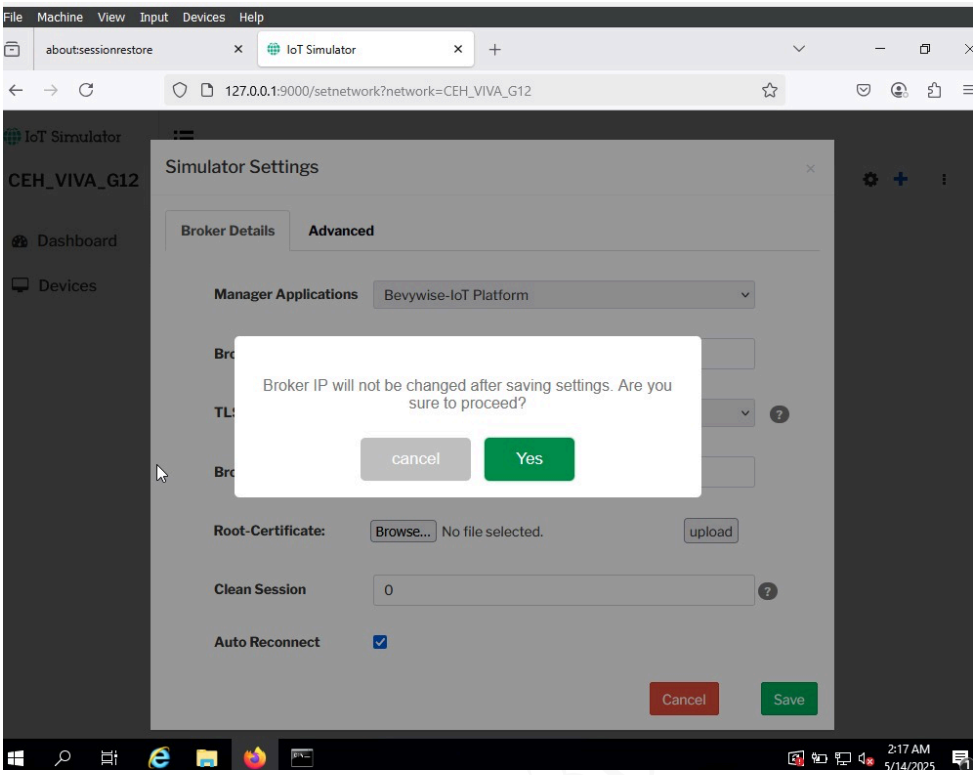


Figure 43 click yes to confirm

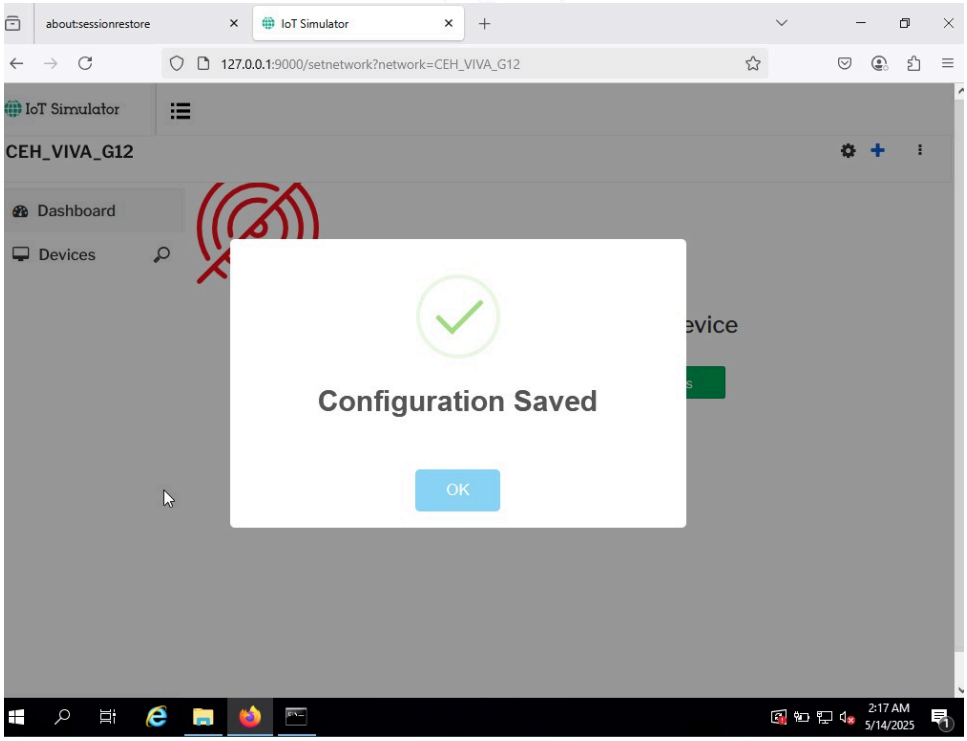


Figure 44 configuration success

Step 9: start network and add a device to your network

Start 9.1: click red dot

Figure 45 click the red dot to start network

Step 9.2 : click the plus and select blank device

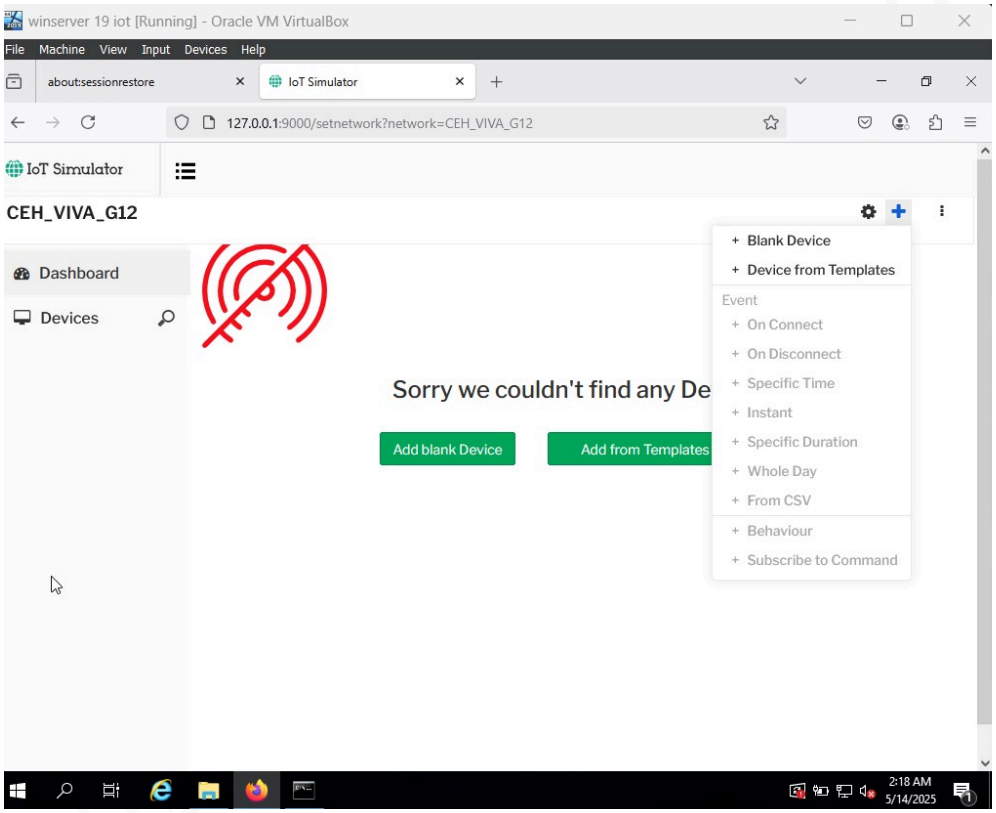


Figure 46 click add blank device or select from drop down menu

Step 9.3: name the device and the device ID (it won't save unless you add a device ID)

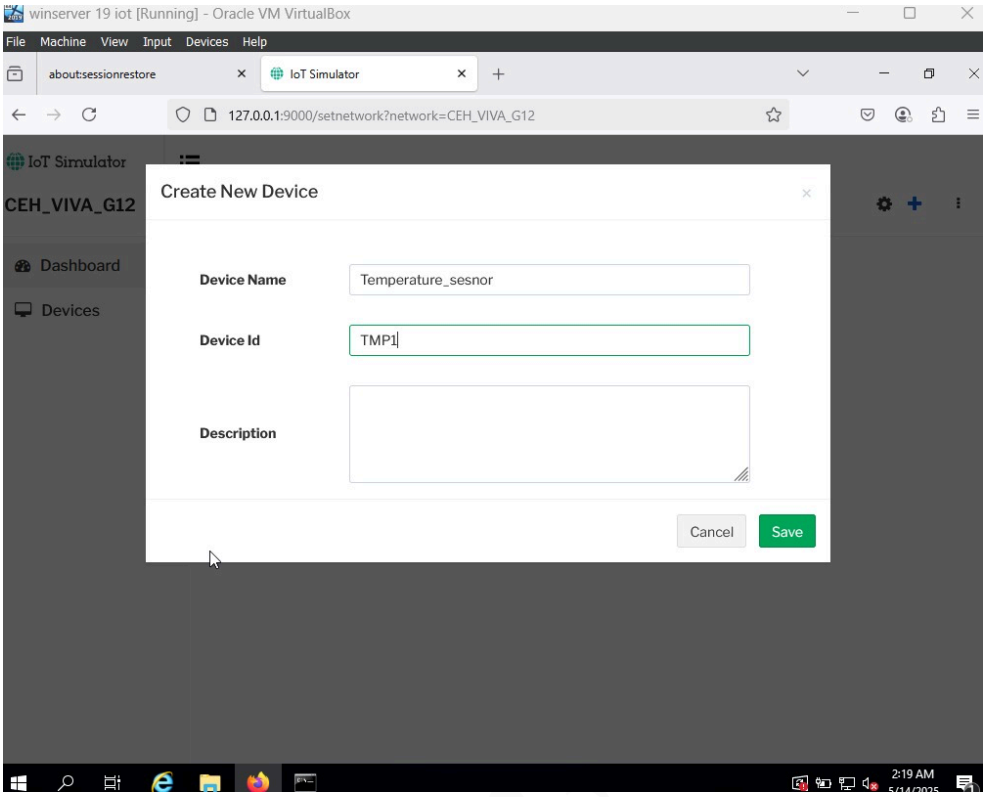


Figure 47 name device and device ID

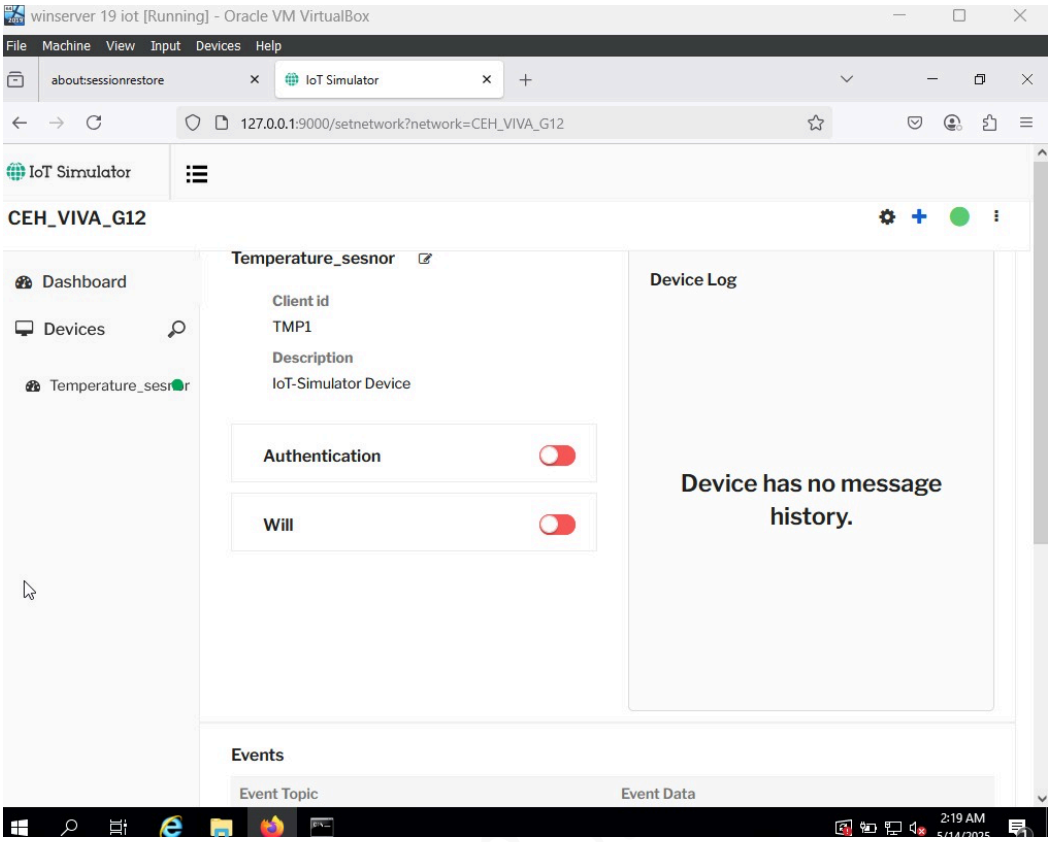


Figure 48 sucesssful device addition

Step 10: Subscribe to command to monitor device communications.

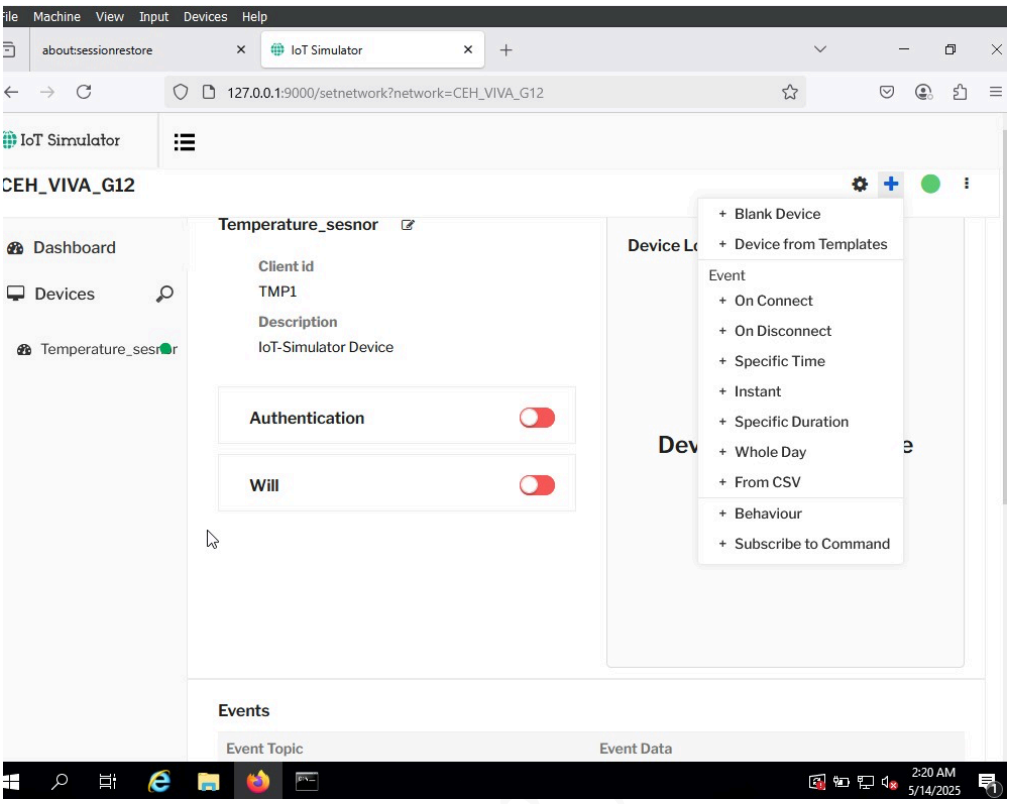


Figure 49 select the subscribe to command option

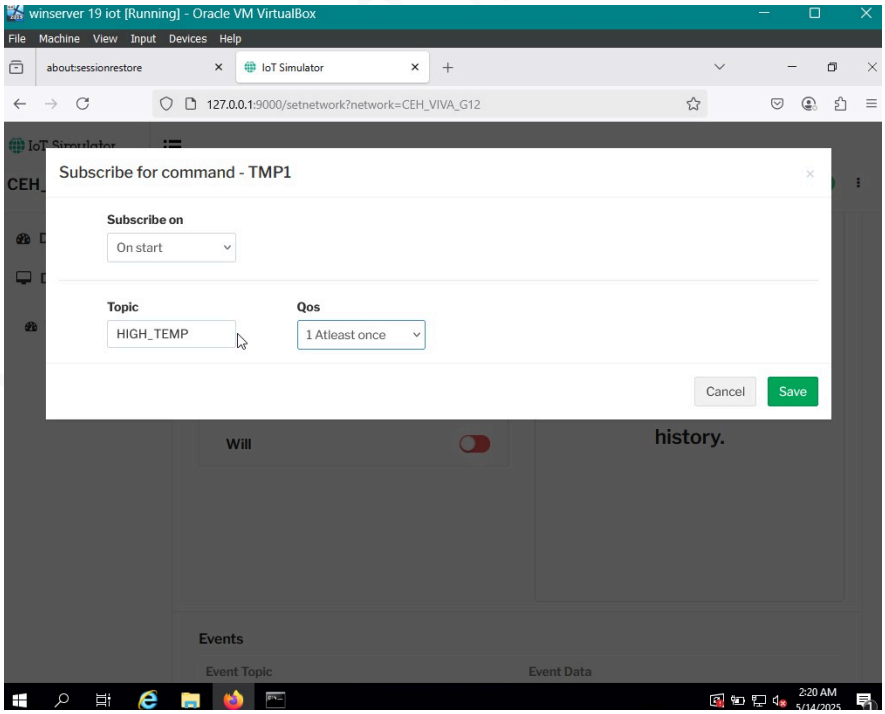


Figure 50 add the details

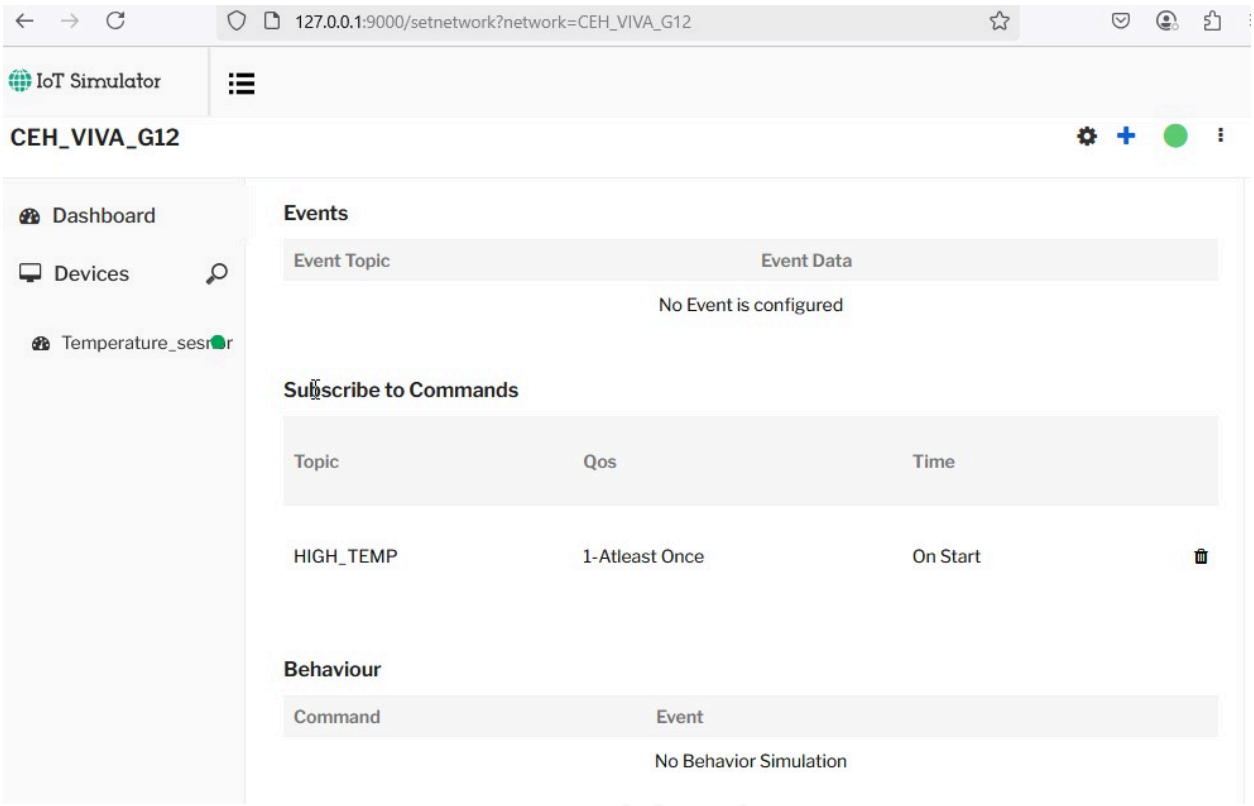
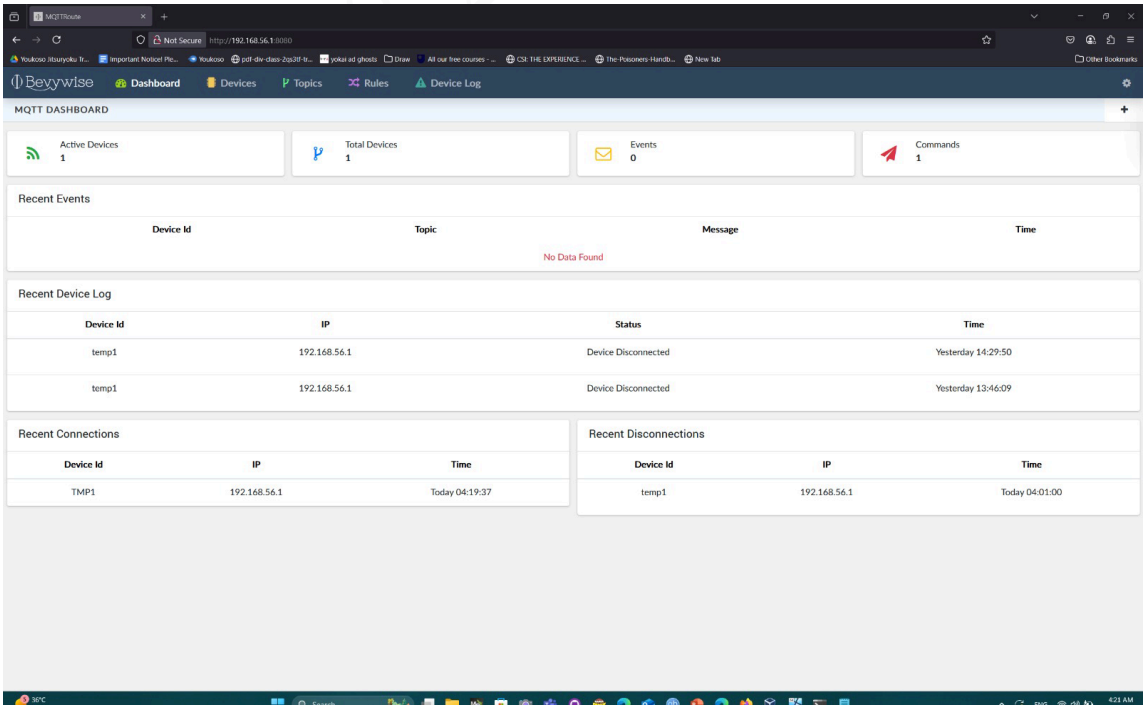


Figure 51 high temp command

Step 11: Check if Device connected and showing on MQTTRoute dashboard



Step 12: In Devices, click the device just set:

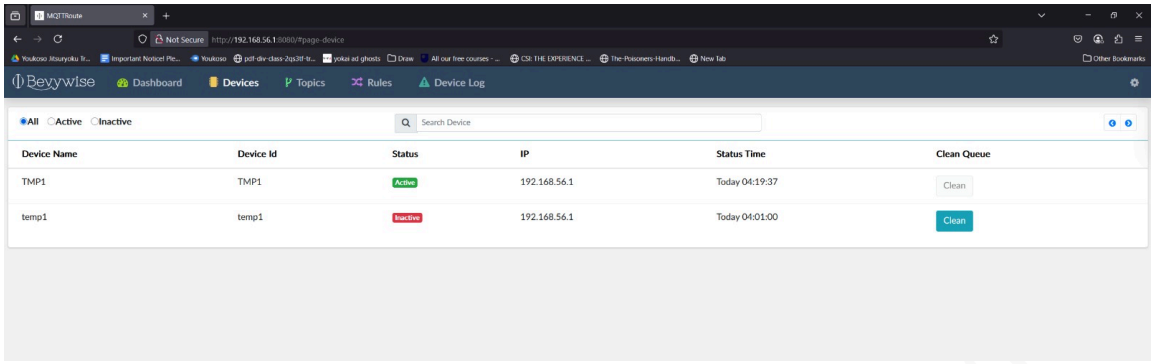


Figure 52 Devices dashboard

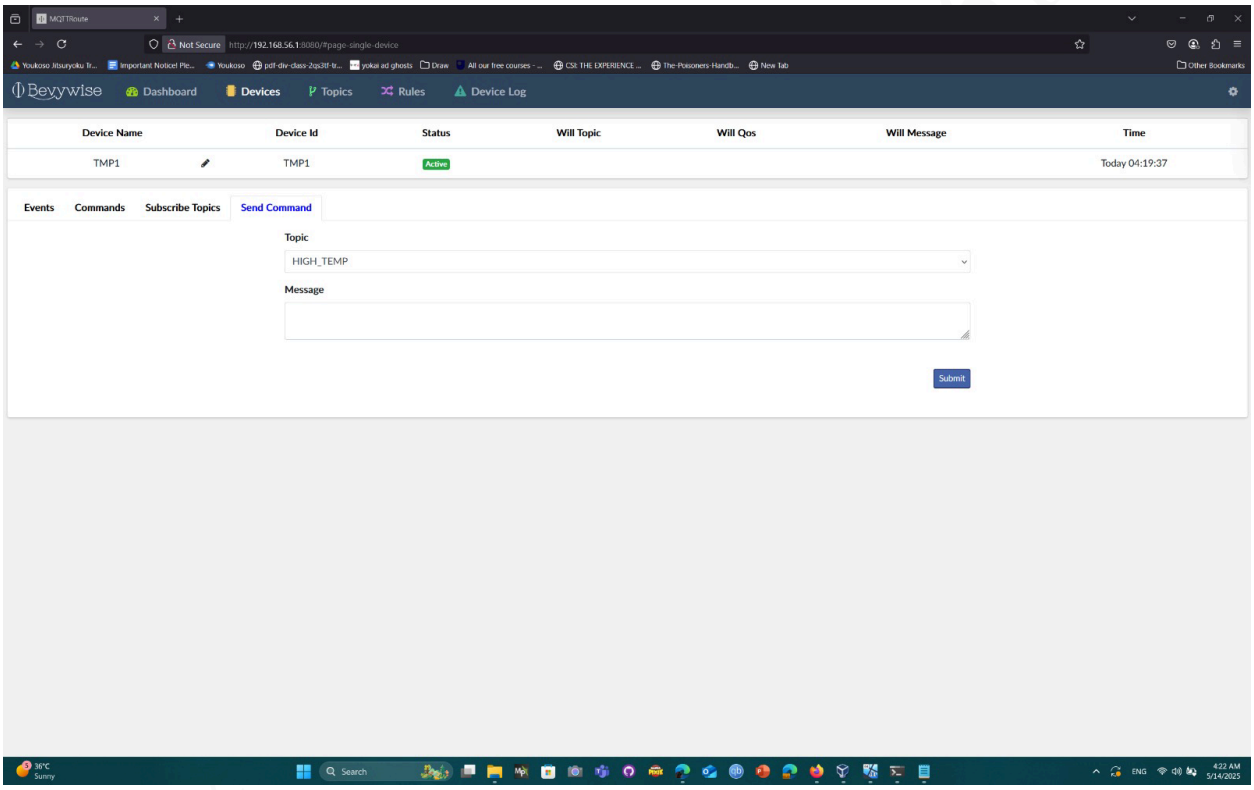


Figure 53 Active device Dashboard

Success you just connected a device to the MQTT broker and verified its status in the web UI.

Step 12: Send a test command:

- **Topic:** HIGH_TEMP
- **Message:** Message sent to TMP1
- Confirmed successful transmission via the web UI.

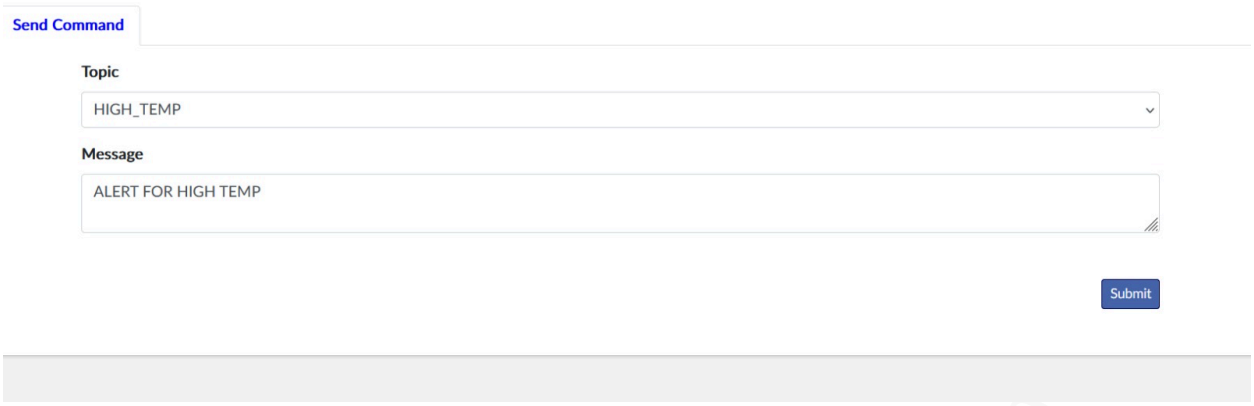


Figure 54 Type out message

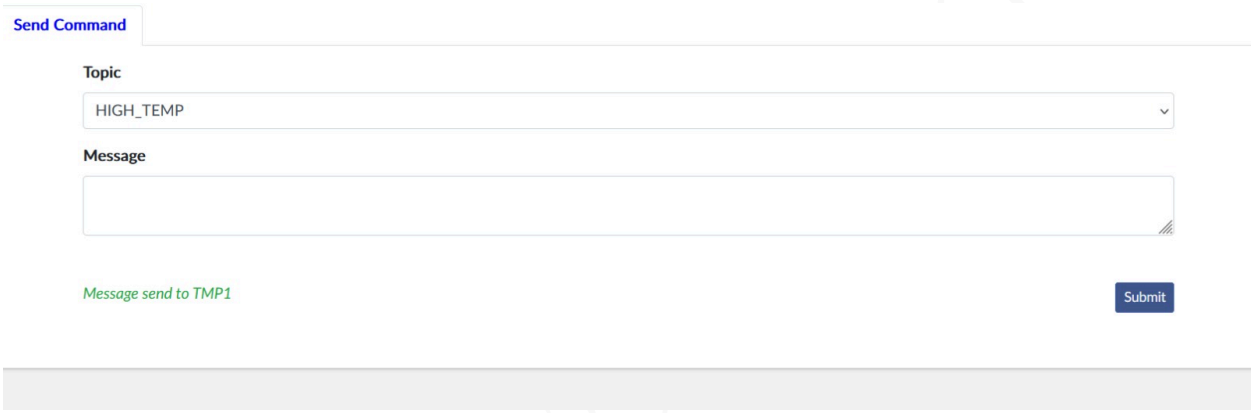


Figure 55 message delivery successful

LAB 3 (NEEDS TO BE IN PARALLEL WITH LAB 2)

Step1: LaunchWireshark and selected the Ethernet interface corresponding to the network hosting the MQTT broker (192.168.56.1) on the windows 2019 server VM

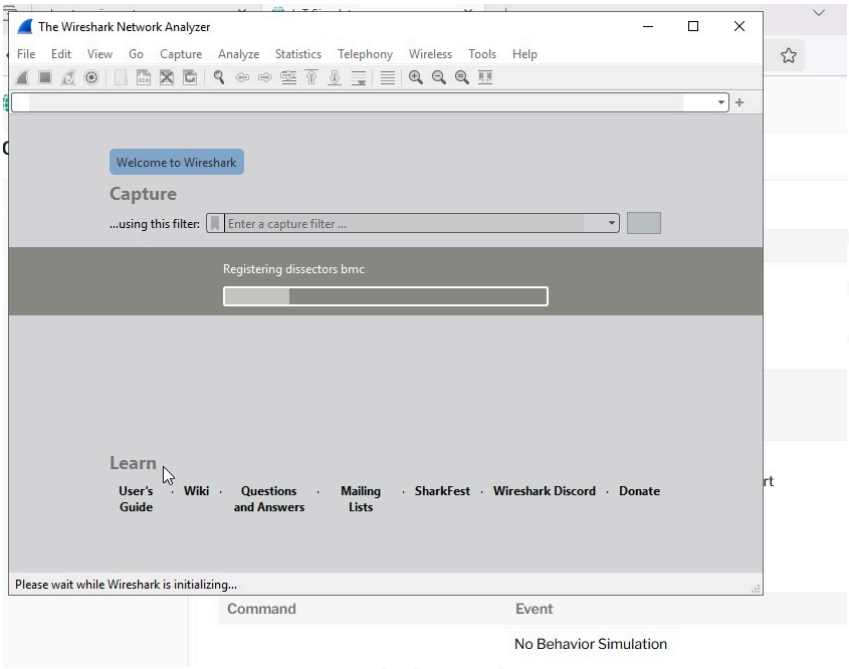


Figure 56 wireshark launched

Step 2: Select Ethernet:

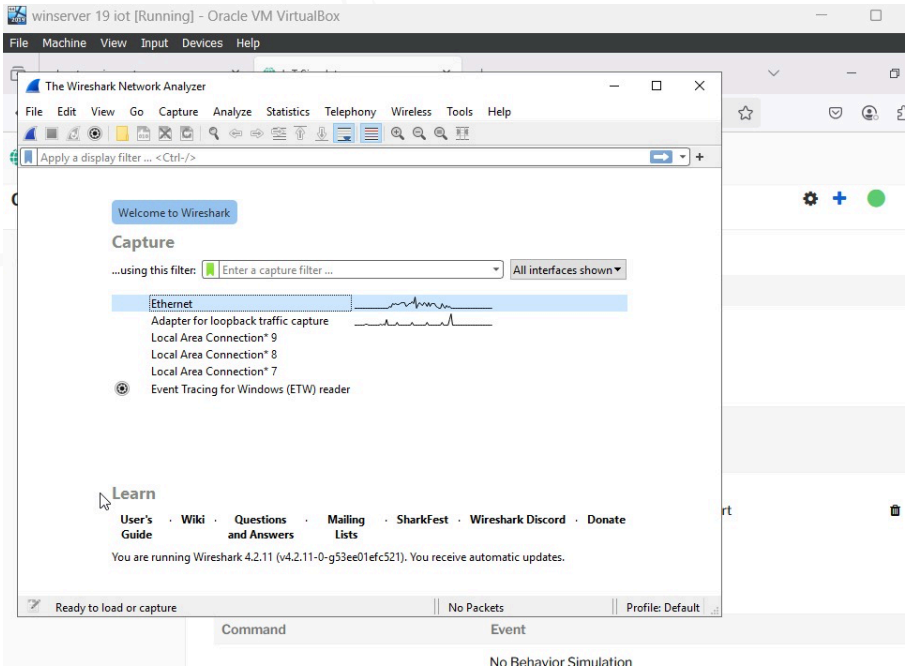


Figure 57 select ethernet

Step 3: Applied a capture filter for the MQTT protocol: TCP port 1883.

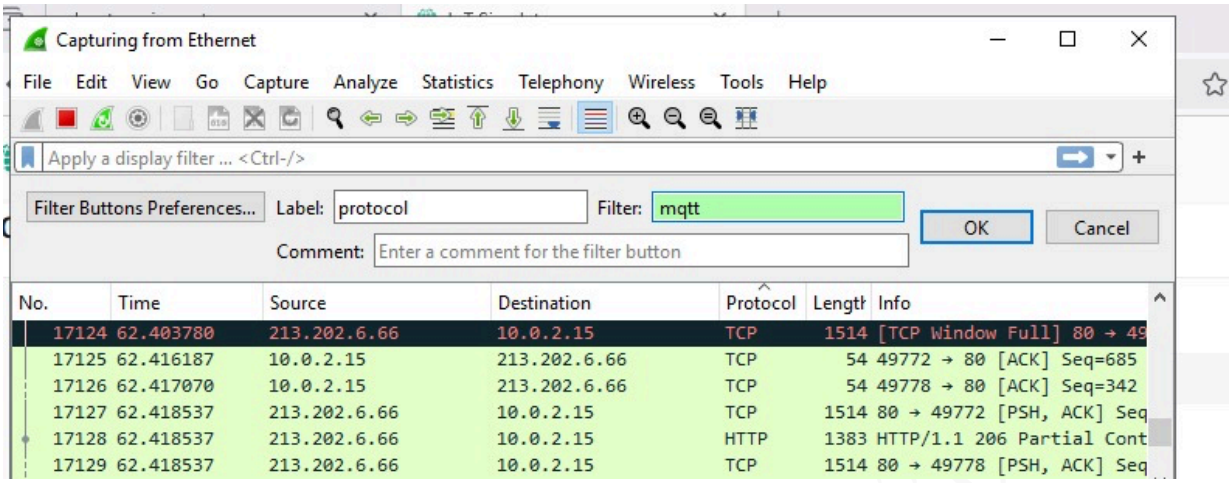


Figure 58 Click the + to create filter

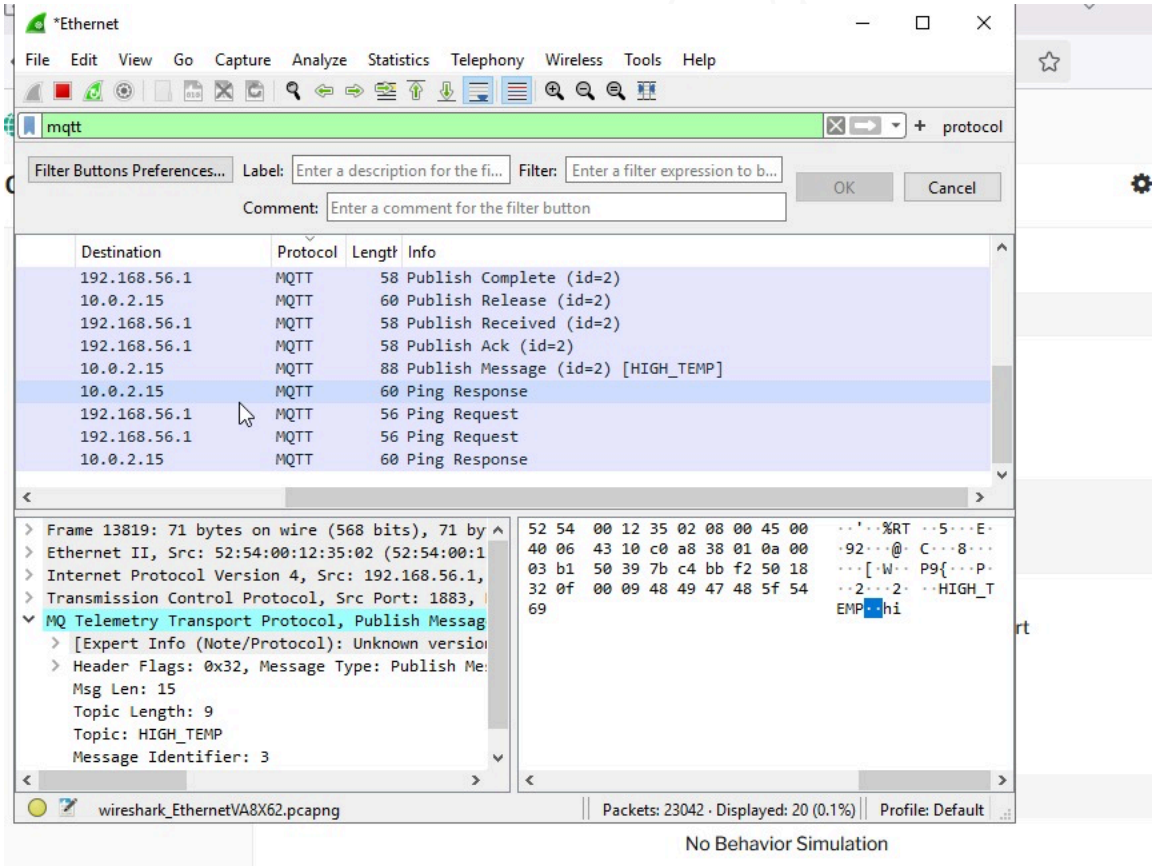


Figure 59 apply the mqtt filter created

Packet Capture Process:

- Initiated packet capture during the MQTT command execution in Lab 2 (e.g., sending the HCHLTEMP command).
- Captured packets included:
    - **Connection Packets:** MQTT CONNECT packets, which included client IDs, protocol versions (e.g., MQTT v3.1.1), and keep-alive timers.
    - **Publish Packets:** MQTT PUBLISH packets containing the topic (HCHLTEMP) and payload (Message sent to TMP1).
    - **Subscription Packets:** MQTT SUBSCRIBE packets indicating the client's subscription to the HCHLTEMP topic.
    - **Acknowledgment Packets:** MQTT CONNACK, PUBACK, and SUBACK packets confirming successful connection, publication, and subscription.

Packet Analysis:

- Analyzed the captured packets to identify:
    - **Unencrypted Data:** The MQTT payload was transmitted in plaintext, exposing the message content (Message sent to TMP1).
    - **Protocol Weaknesses:** Lack of TLS/SSL encryption on port 1883, making the traffic susceptible to interception.
    - **Device Information:** Client IDs and topic names revealed device roles and communication patterns.
- Exported the packet capture for further analysis and documentation.
- Observed packet headers, including source/destination IPs, ports, and Quality of Service (QoS) levels (e.g., QoS 0 for the test command).

ERRORS FACED

Shodan Server Error:

- Encountered intermittent server errors during Shodan searches, likely due to rate-limiting or connectivity issues.
- **Resolution:** Retried searches after a brief delay and verified account credentials.

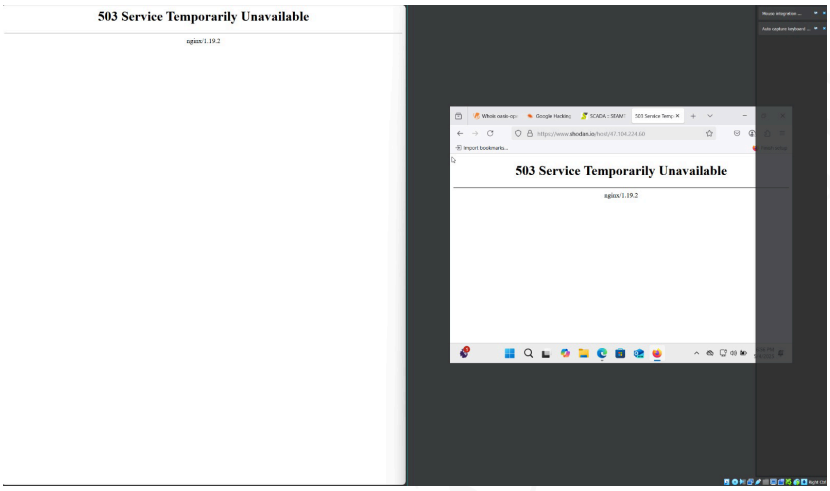


Figure 60 shodan down ...

Window Error:

Solution delete the file attached to the floppydisk when entering the virtual box vm storage settings.

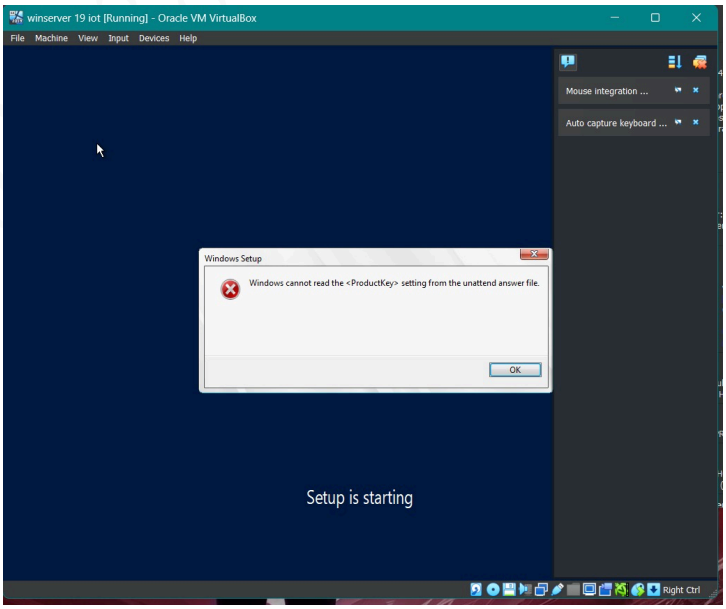


Figure 61 windows startup error loop

MQTTRoute Errors:

- **Device Not Visible on Web UI:** Received an error stating, “Broker is not running. Start broker to connect the devices.”
    - **Resolution:** Restarted the MQTTRoute service and verified the broker’s status in the terminal.
- **Subscription Command Failed:** Subscription attempts failed due to misconfigured topic names or firewall restrictions.
    - **Resolution:** Corrected topic names and adjusted firewall settings (see below).

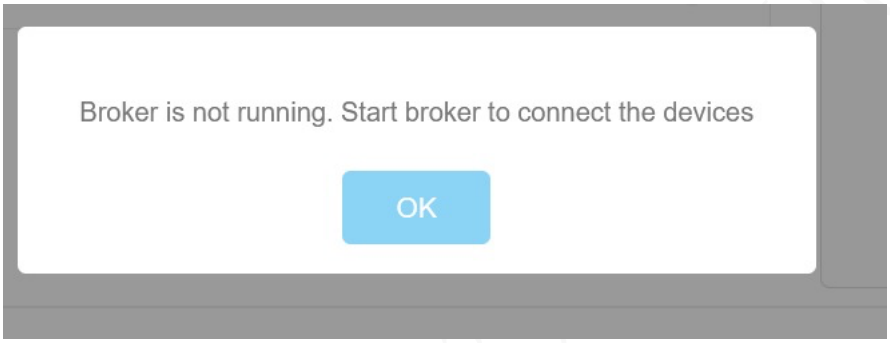


Figure 62 network not staring for the iot network

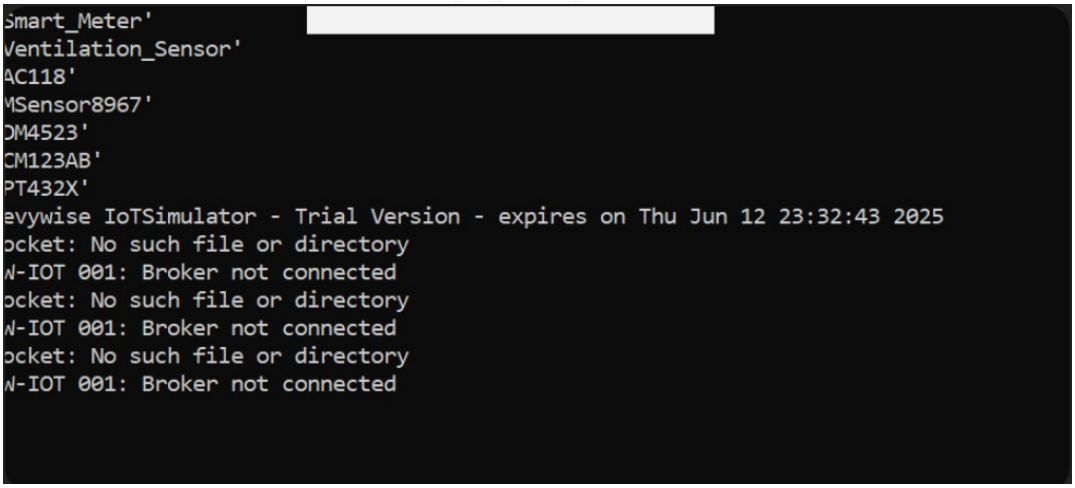


Figure 63 broker not foun in the iot simulator terminal

Firewall Configuration Issue:

- MQTT traffic on port 1883 was blocked by Windows Defender Firewall.
- **Resolution:** Created a new inbound rule in Windows Defender Firewall:
    - **Protocol:** TCP

- **Port:** 1883
- **Action:** Allow the connection
- **Profile:** Applied to Domain, Private, and Public networks
- **Name:** MQTT_PORT

- Verified that the rule resolved connectivity issues for MQTT communications.

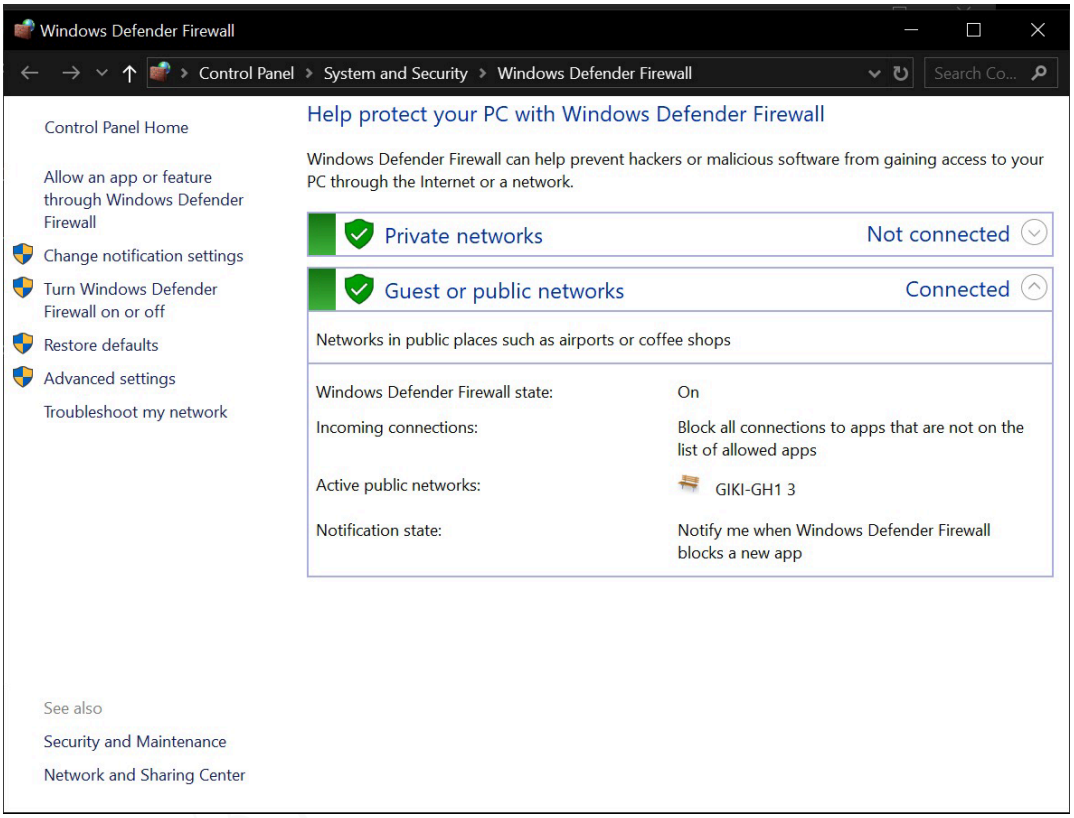


Figure 64 firewall interface

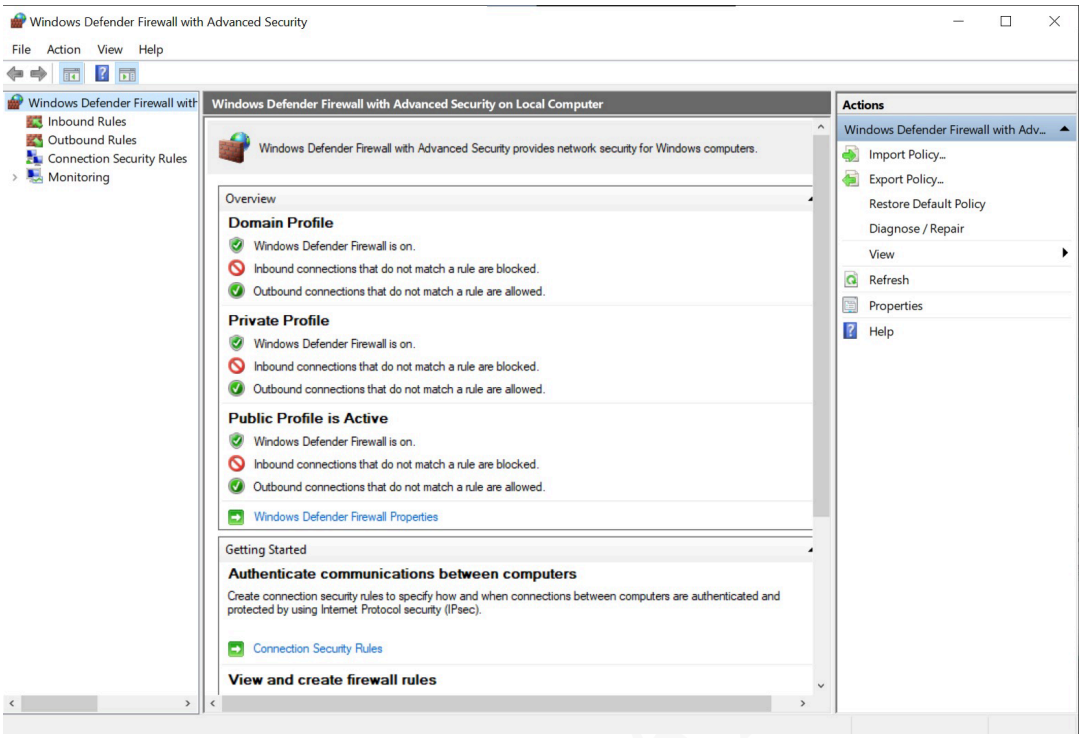


Figure 65 windows defender firewall overview

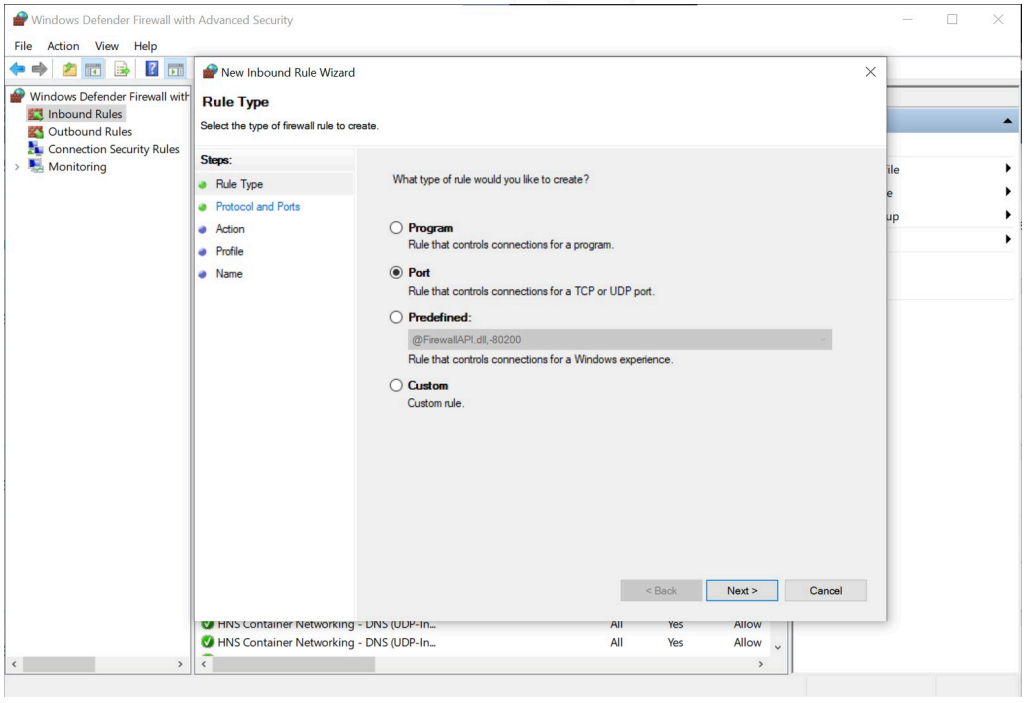


Figure 66 add inbound rules to open port 1883

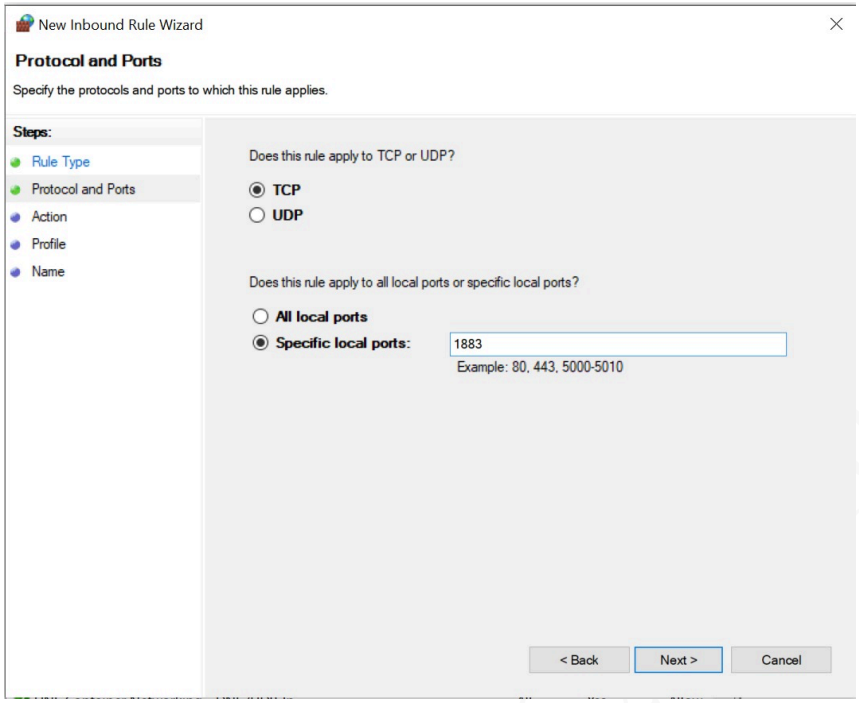


Figure 67 click next

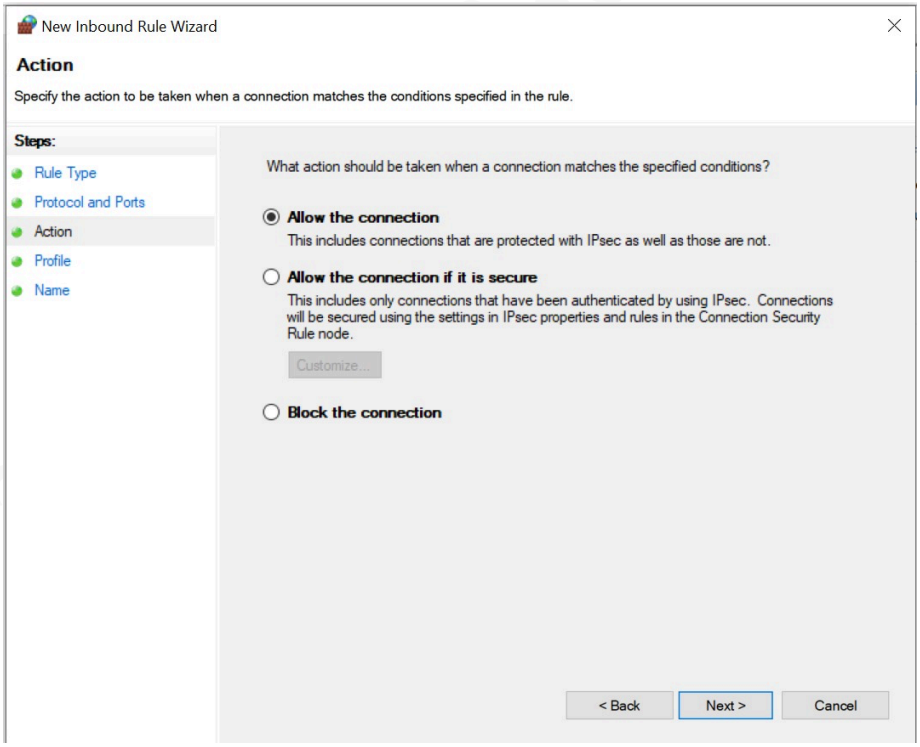


Figure 68 click next

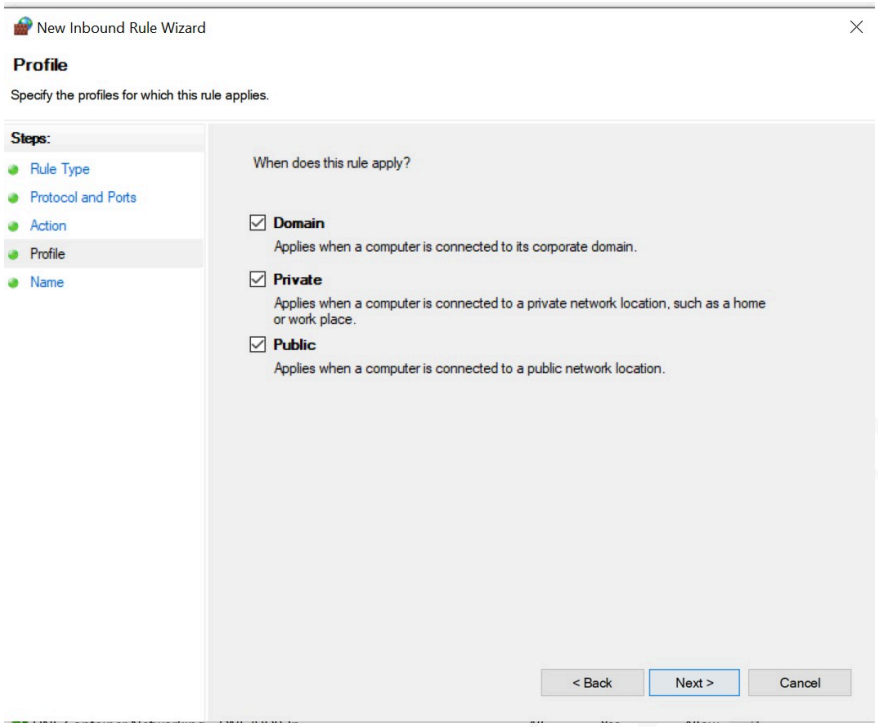


Figure 69 click next

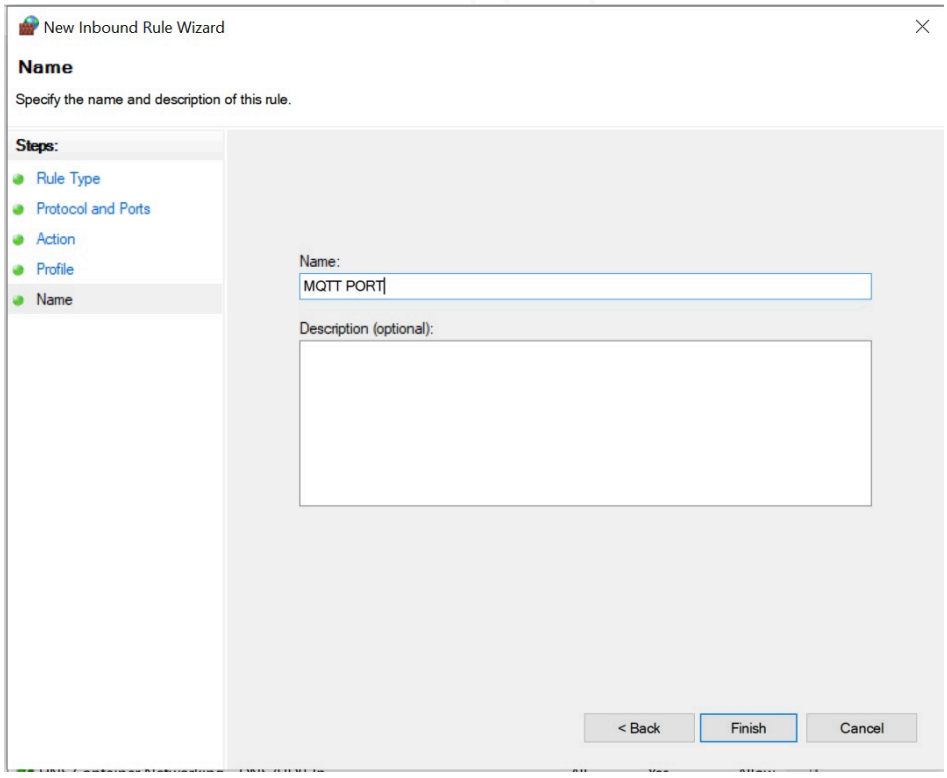


Figure 70 name the rule

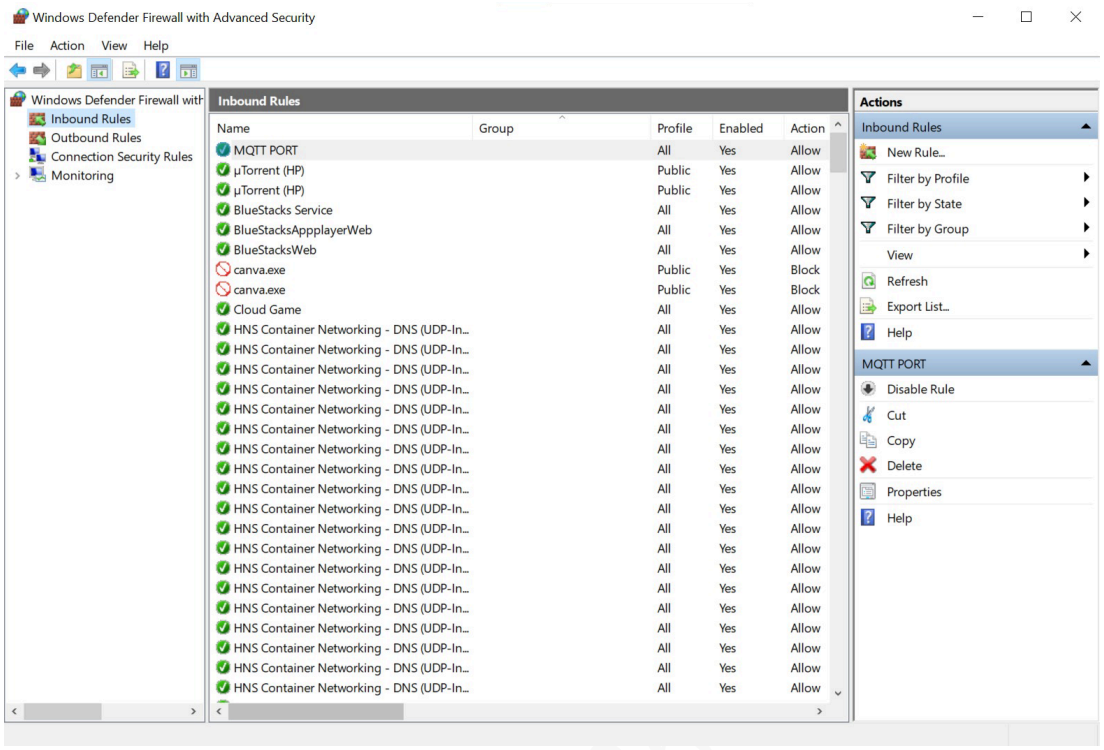


Figure 71 check the rules for the new rule to verify

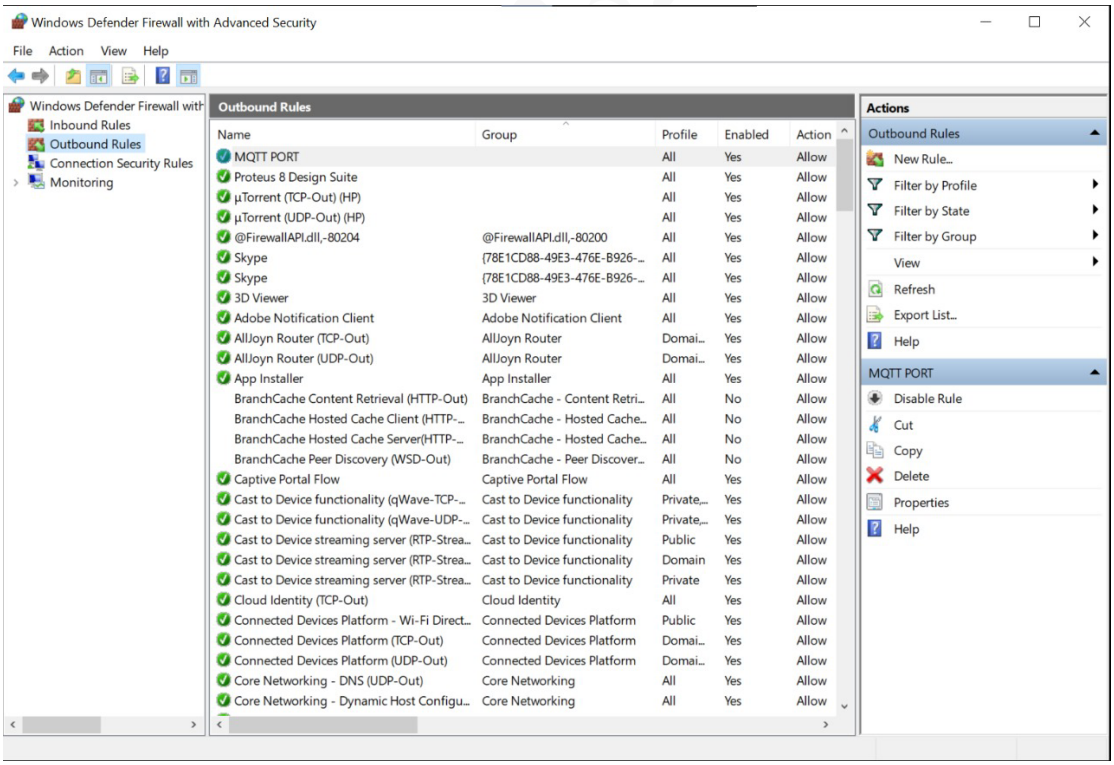


Figure 72 do the same and add an outbound rule

FINAL LEARNING

The IoT and OT Hacking lab provided hands-on experience in ethical hacking and security analysis of IoT and OT systems. Key takeaways include:

- **Footprinting Proficiency: Mastered tools like Shodan and Whois for gathering intelligence on IoT devices, including IP addresses, open ports, and service details. Google dorking revealed exposed SCADA systems, highlighting the risks of misconfigured interfaces.**
- **Traffic Analysis Expertise: Developed advanced skills in using Wireshark to capture and analyze MQTT traffic. The ability to dissect packets (e.g., CONNECT, PUBLISH, SUBSCRIBE) revealed critical vulnerabilities, such as unencrypted payloads and weak authentication.**
- **Vulnerability Identification: Identified security weaknesses in IoT communications, including plaintext MQTT traffic and brute-force vulnerabilities in SCADA login portals.**
- **Practical Attack Simulation: Simulated real-world attack scenarios by intercepting MQTT commands and analyzing device interactions, reinforcing the importance of secure protocol configurations.**
- **Problem-Solving Skills: Overcame technical challenges (e.g., firewall issues, broker errors) through systematic troubleshooting and configuration adjustments.**
- **Security Awareness: Gained a deeper understanding of securing IoT ecosystems by implementing firewall rules and recognizing the need for encryption (e.g., MQTT over TLS).**

These skills have strengthened our ability to secure IoT and OT environments against cyber threats and prepared us for real-world cybersecurity challenges.