

IoT and OT Hacking

Module 18

IoT and OT Hacking

IoT and OT device hacking is performed to compromise smart devices such as CCTV cameras, automobiles, printers, door locks, washing machines, etc. to gain unauthorized access to network resources as well as IoT and OT devices.

Lab Scenario

ICON KEY



Valuable information



Test your knowledge



Web exercise



Workbook review

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.

Lab Objectives


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

Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Ubuntu virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

 **Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 18 IoT and OT Hacking**

Lab Duration

Time: 30 Minutes

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:

Lab No.	Lab Exercise Name	Core*	Self-study**	iLabs ***
1	Perform Footprinting using Various Footprinting Techniques	√		√
	1.1 Gather Information using Online Footprinting Tools	√		√
2	Capture and Analyze IoT Device Traffic	√		√
	2.1 Capture and Analyze IoT Traffic using Wireshark	√		√

Remark

EC-Council has prepared a considered amount of lab exercises for student to practice during the 5-day class and at their free time to enhance their knowledge and skill.

***Core** - Lab exercise(s) marked under Core are recommended by EC-Council to be practised during the 5-day class.

****Self-study** - Lab exercise(s) marked under self-study is for students to practise at their free time. Steps to access the additional lab exercises can be found in the first page of CEHv11 volume 1 book.

*****iLabs** - Lab exercise(s) marked under iLabs are available in our iLabs solution. iLabs is a cloud-based virtual lab environment preconfigured with vulnerabilities, exploits, tools and scripts, and can be accessed from anywhere with an Internet connection. If you are interested to learn more about our iLabs solution, please contact your training center or visit <https://ilabs.eccouncil.org>.

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.





PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS
RELATED TO THIS LAB.



Perform Footprinting using Various Footprinting Techniques

Ethical hackers and penetration testers are aided in footprinting by various tools that make information gathering an easy task.

ICON KEY

-  Valuable Information
-  Test Your Knowledge
-  Web Exercise
-  Workbook Review

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

Lab Environment

To carry out this lab, you need:


- Windows 10 virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 15 Minutes

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

 **Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 18 IoT and OT Hacking**

includes IP address, hostname, ISP, device location, banner of the target IoT device, FCC ID information, certification granted to the device, etc.

Lab Tasks



TASK 1

Gather Information using Online Footprinting Tools

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. Turn on the **Windows 10** virtual machine and login with the credentials **Admin** and **Pa\$\$w0rd**.
2. Open any web browser (here, **Mozilla Firefox**), type **https://www.whois.com/whois/** in the address bar, and press **Enter**.
3. 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.



TASK 1.1

Perform Whois Footprinting

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.



Figure 1.1.1: Whois Domain Lookup page

- 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.

The screenshot shows a web browser window with the title 'Whois oasis-open.org'. The address bar displays 'https://www.whois.com/' with a zoom level of 80%. The main content area shows the results for 'oasis-open.org', which was updated 1 second ago. The results are divided into two sections: 'Domain Information' and 'Registrant Contact'. The 'Domain Information' section lists the domain, registrar, registration and expiration dates, update date, status, and name servers. The 'Registrant Contact' section lists the organization, state, and country.

Domain Information	
Domain:	oasis-open.org
Registrar:	DNC Holdings, Inc.
Registered On:	1998-03-04
Expires On:	2021-03-03
Updated On:	2020-01-18
Status:	clientDeleteProhibited clientTransferProhibited clientUpdateProhibited
Name Servers:	dns2.stabletransit.com dns1.stabletransit.com

Registrant Contact	
Organization:	OASIS Open
State:	MA
Country:	US

Figure 1.1.2: Whois lookup result: Domain Information, Registrant Contact

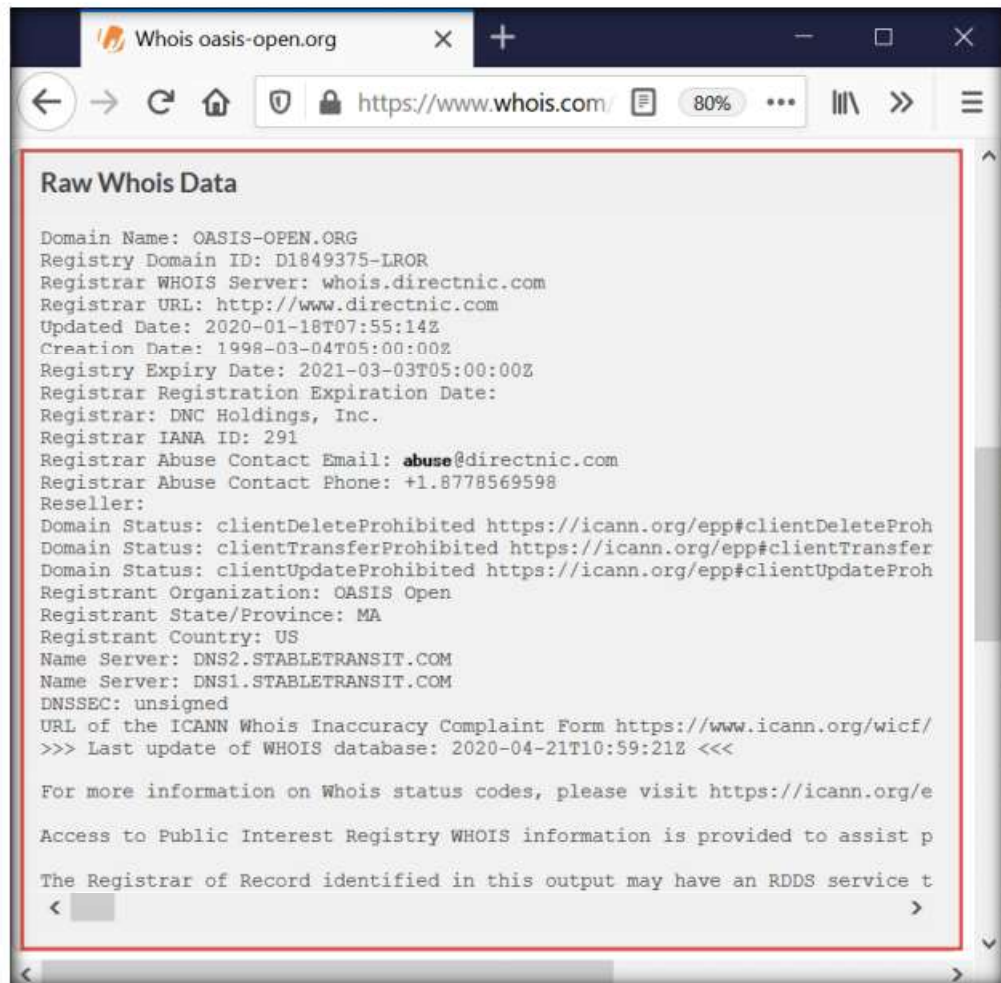


Figure 1.1.3: Whois lookup result: Raw Whois Data

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

- Now, press **Ctrl+T** to open a new tab, type **https://www.exploit-db.com/google-hacking-database** in the address bar, and press **Enter**.
- The **Google Hacking Database** page appears; type **SCADA** in the **Quick Search** field and press **Enter**.
- The result appears, which displays the Google dork related to SCADA, as shown in the screenshot.



TASK 1.2

Perform Advanced Google Hacking

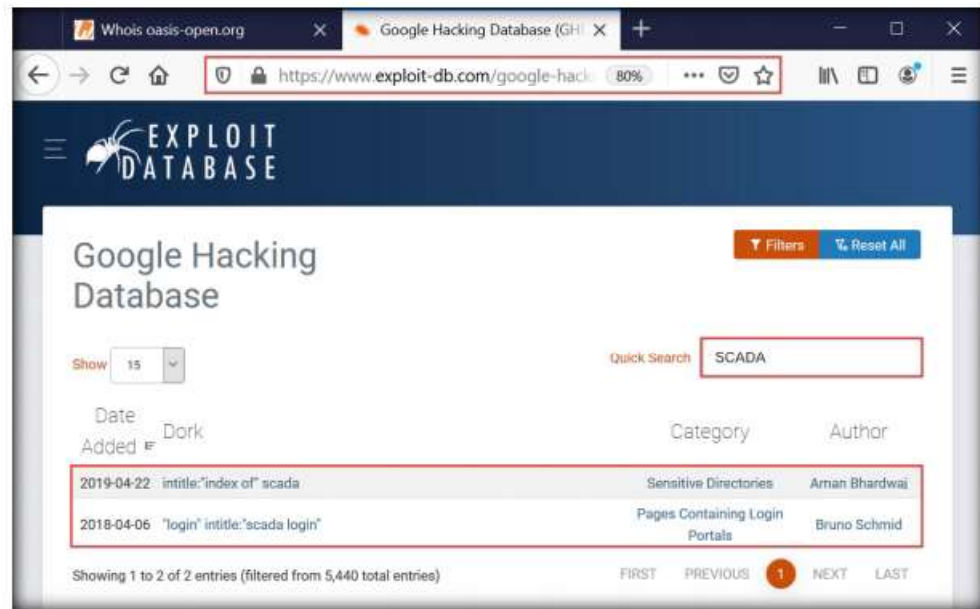


Figure 1.1.4: Google Hacking Database result

8. Now, we will use the dorks obtained in the previous step to query results in Google.
9. Press **Ctrl+T** to open a new tab, type **https://www.google.com** in the address bar, and press **Enter**.
10. In the search field, type **"login" intitle:"scada login"** and click the **Google Search** button.

Note: By default, the tool is cloned to the root directory.

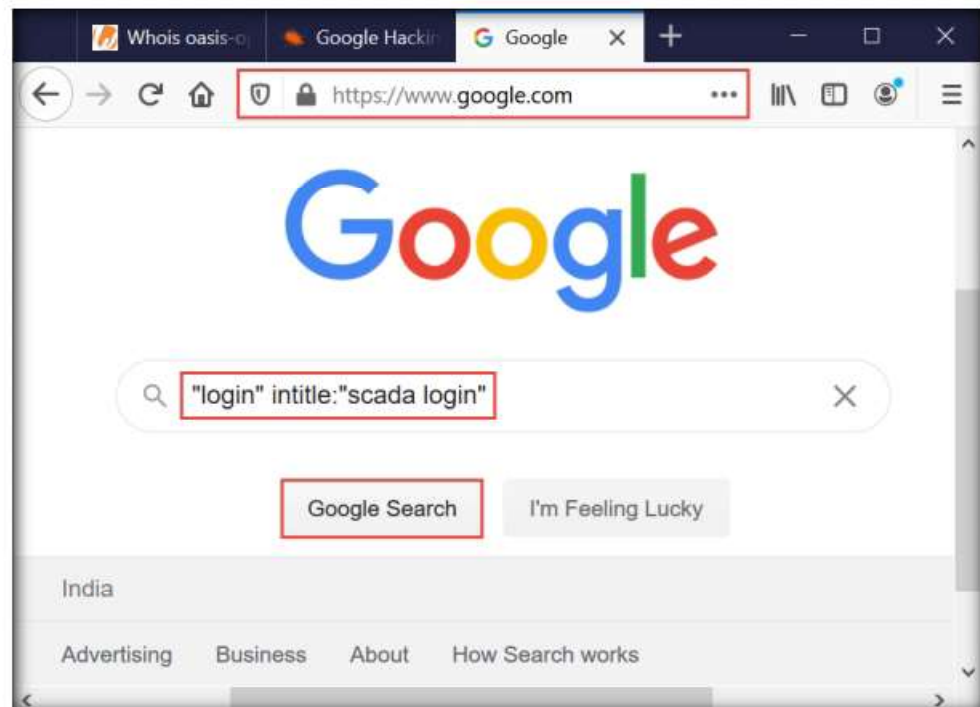


Figure 1.1.5: Navigating to the cloned folder and listing the folder content

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

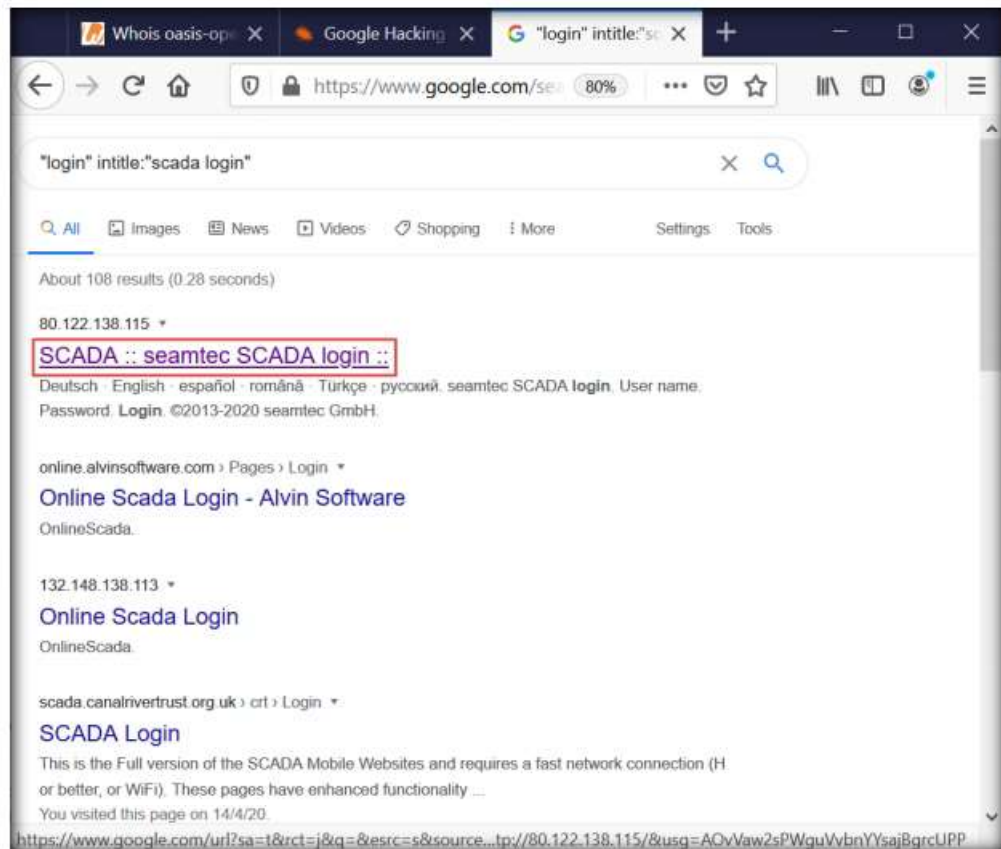


Figure 1.1.6: SCADA login page search result

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.

12. 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.

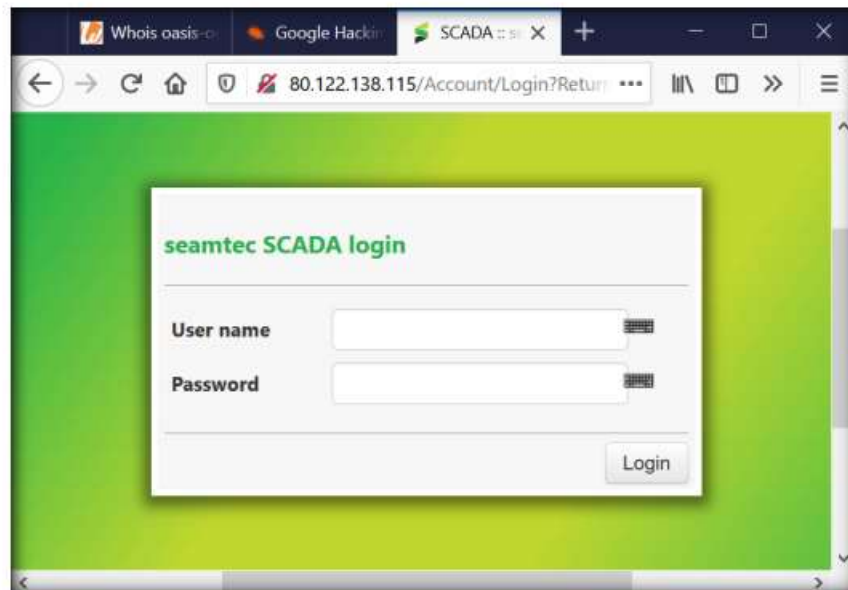


Figure 1.1.7: Seamtec SCADA login form

TASK 1.3

Perform Footprinting using Shodan

13. Similarly, you can use advanced search operators such as **intitle:"index of" scada** to search sensitive SCADA directories that are exposed on sites.
14. Now, in the browser window, press **Ctrl+T** to open a new tab, type **https://account.shodan.io/login** in the address bar, and press **Enter**.
15. 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.

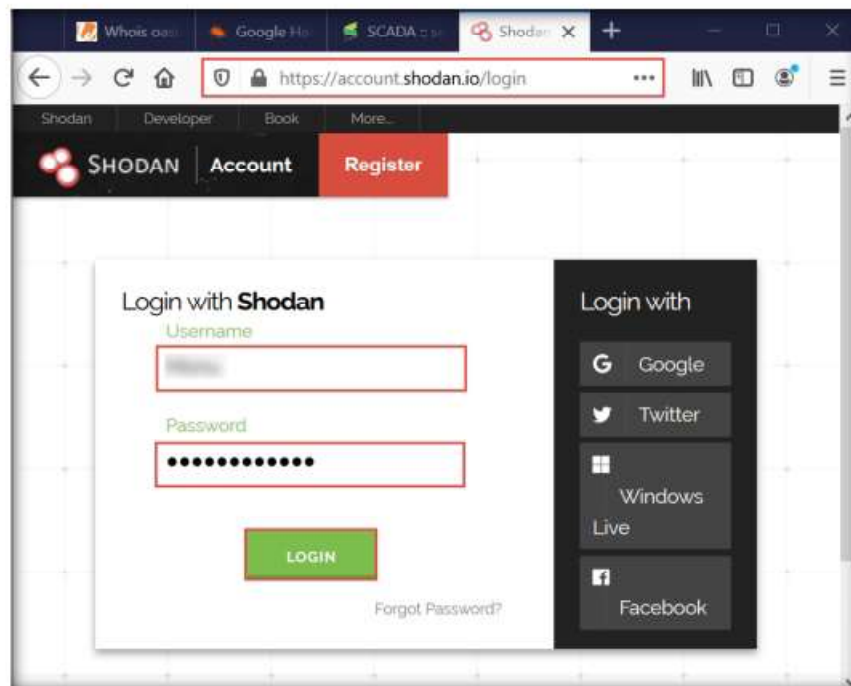


Figure 1.1.8: Login with Shodan page

16. 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**.

17. Click **Shodan** in the top-left corner of the window.

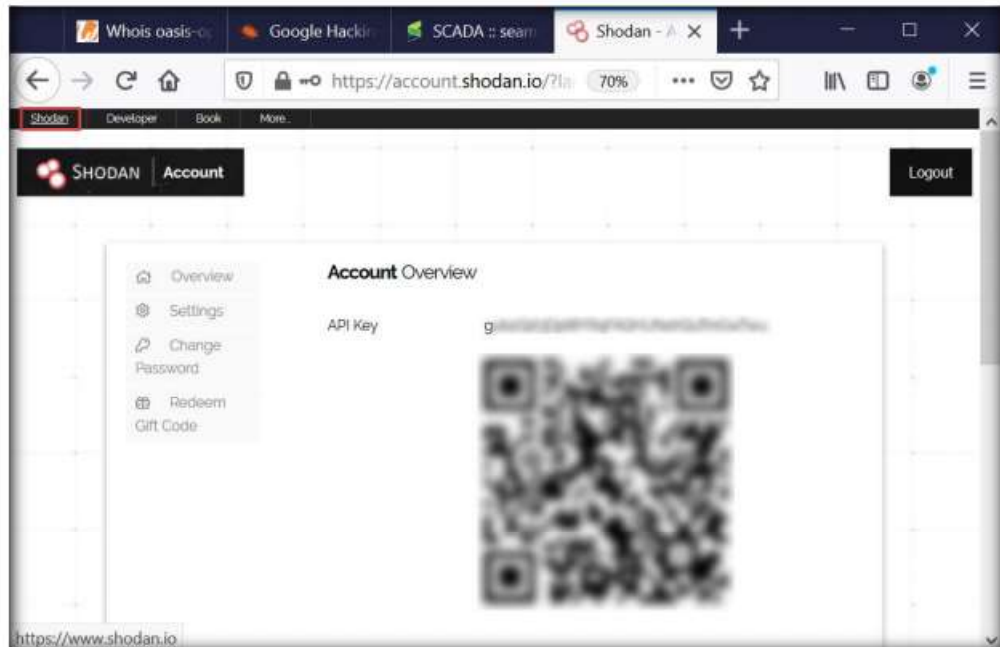


Figure 1.1.9: Account Overview page

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.



Figure 1.1.10: Shodan main page

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.

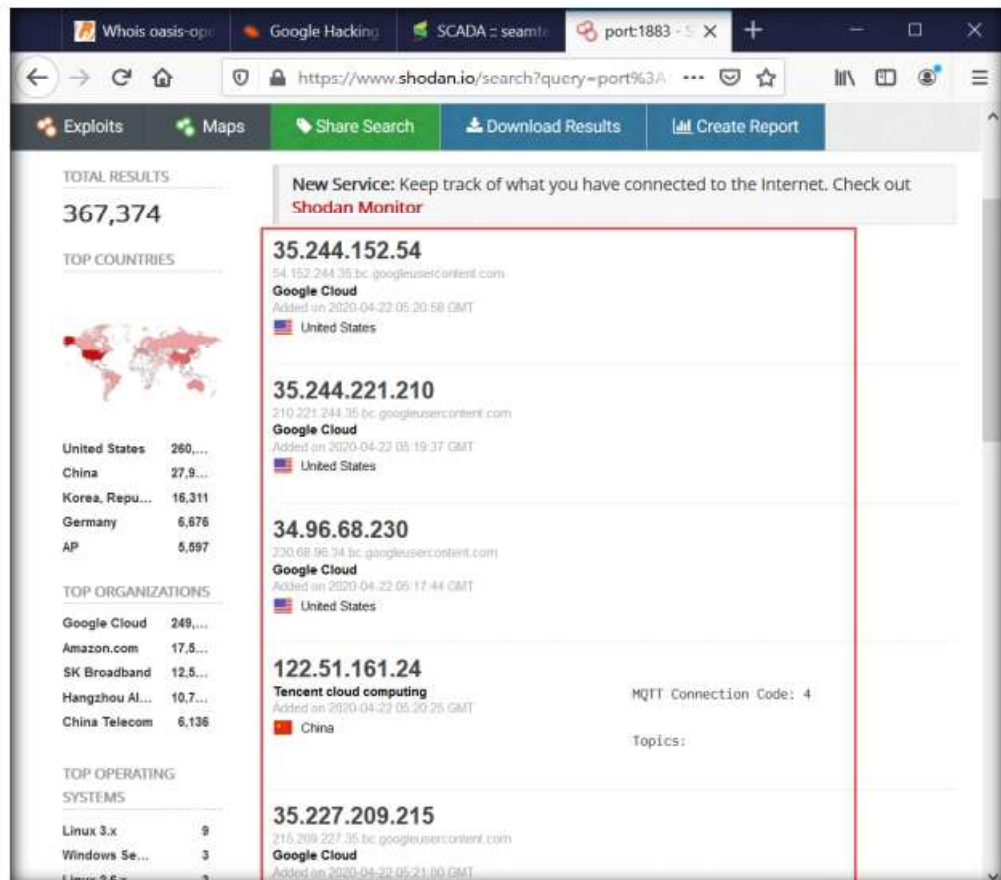


Figure 1.1.11: Port 1883 result page

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

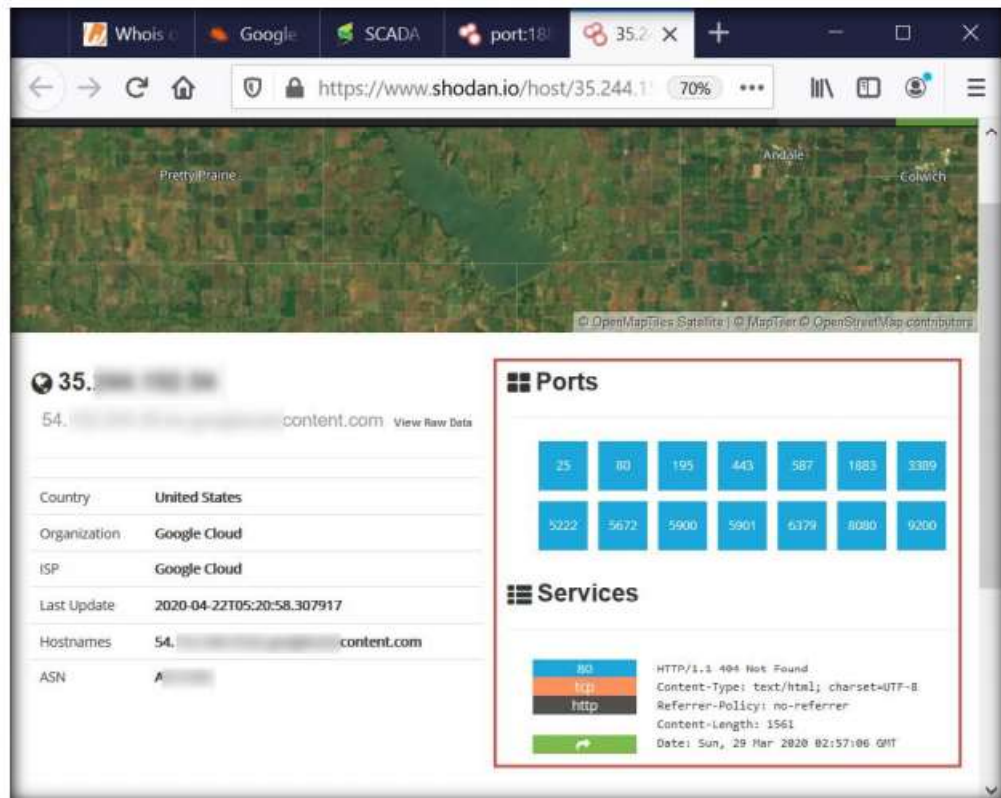


Figure 1.1.12: Detailed information

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.
26. Turn off the **Windows 10** virtual machine.

Lab Analysis

Analyze and document all the results obtained in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS
ABOUT THIS LAB.


Internet Connection Required	
<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
Platform Supported	
<input checked="" type="checkbox"/> Classroom	<input checked="" type="checkbox"/> iLabs


Lab 2

Capture and Analyze IoT Device Traffic


IoT refers to a network of devices having an IP address as well as the capability to sense, collect, and send data using embedded sensors, communication hardware, and processors.

ICON KEY

 Valuable Information

 Test Your Knowledge

 Web Exercise

 Workbook Review

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

Lab Environment

To carry out this lab, you need:


- Windows 10 virtual machine
- Ubuntu virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 15 Minutes

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,

 **Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 18 IoT and OT Hacking**

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.

Lab Tasks



TASK 1

Capture and Analyze IoT Traffic using Wireshark

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

1. Turn on the **Windows 10** and **Ubuntu** virtual machines.
2. In the **Ubuntu** virtual machine, click on the **Ubuntu** button, type **toor** in the **Password** field, and press **Enter** to sign in to the machine.

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.

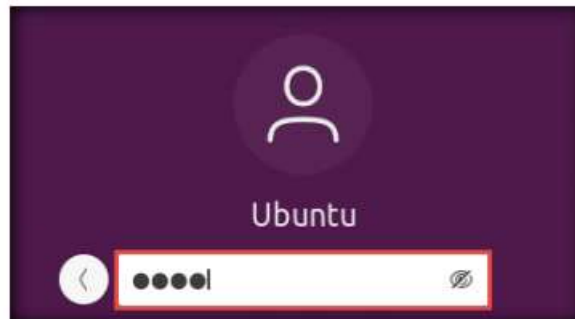


Figure 2.1.1: Ubuntu login


3. In the left pane under the **Activities** list, scroll down and click the **Terminal** () icon to open the terminal window.



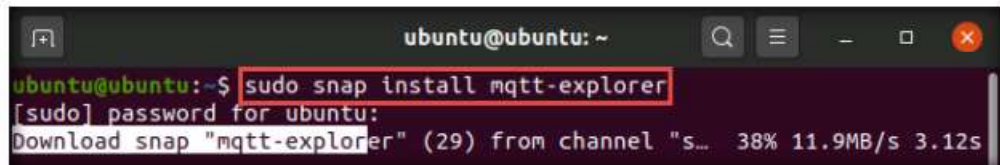
Figure 2.1.2: Open Terminal window

TASK 1.1**Install
MQTT Explorer**

- In the terminal window, type **sudo snap install mqtt-explorer** and press **Enter** to install the MQTT Explorer tool.

Note: MQTT Explorer is a comprehensive MQTT client that provides a structured overview of your MQTT topics and simplifies working with devices/services on your broker.

- In the **[sudo] password for ubuntu** option, enter **toor** as the password and press **Enter**.

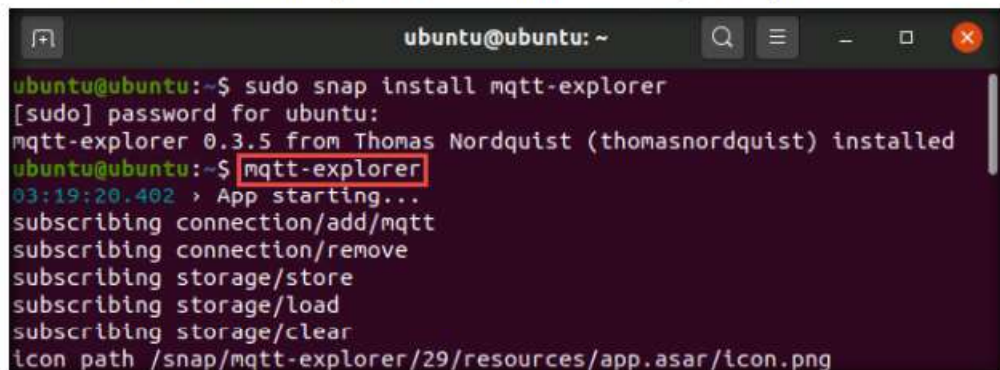


```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sudo snap install mqtt-explorer
[sudo] password for ubuntu:
Download snap "mqtt-explorer" (29) from channel "s... 38% 11.9MB/s 3.12s
```

Figure 2.1.3: Installing MQTT Explorer

TASK 1.2**Launch
Wireshark**

- After the installation is complete, switch to the **Windows 10** virtual machine and login with the credentials **Admin/Pa\$\$w0rd**.
- Open the **Wireshark** application and double-click the available Ethernet or interface (here, **Ethernet0**) to start capturing the packets.
- Leave the Wireshark window running.
- Switch back to the **Ubuntu** virtual machine. In the terminal window, type **mqtt-explorer** and press **Enter** to launch the MQTT Explorer tool.



```
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sudo snap install mqtt-explorer
[sudo] password for ubuntu:
mqtt-explorer 0.3.5 from Thomas Nordquist (thomasnordquist) installed
ubuntu@ubuntu:~$ mqtt-explorer
03:19:20.402 > App starting...
subscribing connection/add/mqtt
subscribing connection/remove
subscribing storage/store
subscribing storage/load
subscribing storage/clear
icon path /snap/mqtt-explorer/29/resources/app.asar/icon.png
```

Figure 2.1.4: AWS CLI Installed Successfully

TASK 1.3**Launch MQTT
Explorer**

- The MQTT Explorer tool initializes and the **MQTT Explorer** main window appears, as shown in the screenshot.
- In the **MQTT Connection** window, click **CONNECT**.

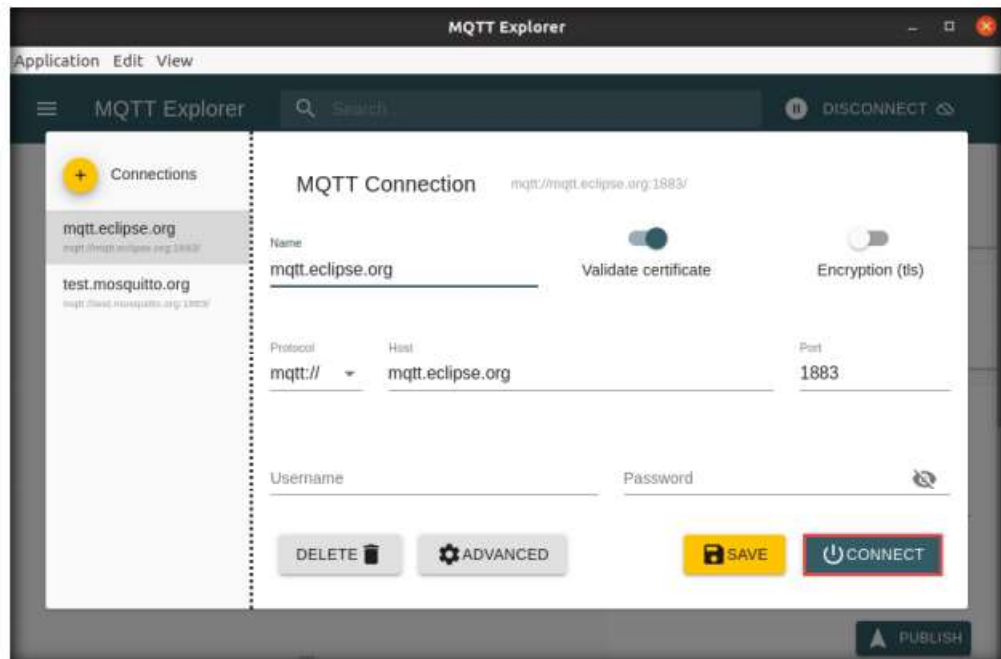


Figure 2.1.5: MQTT Explorer window

12. **MQTT Explorer** starts establishing a connection with the devices mentioned in the left pane, as shown in the screenshot.

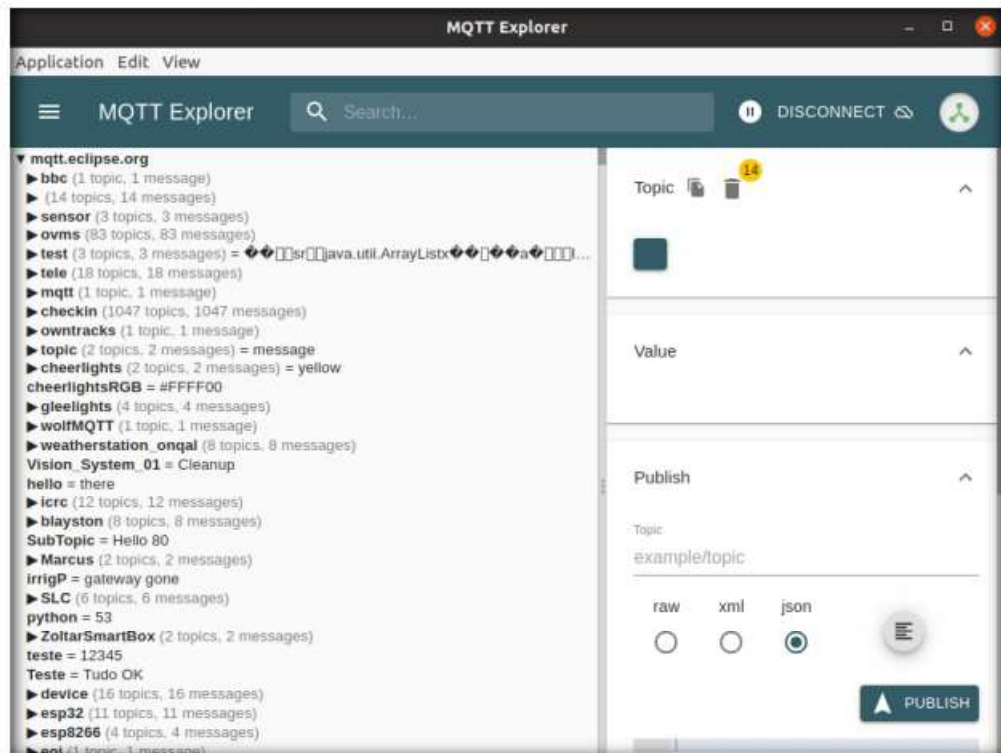


Figure 2.1.6: MQTT Explorer connecting with the devices

13. Wait for some time, and then click **DISCONNECT** in the top section of the **MQTT Explorer** window to disconnect the tool.
14. Switch to the **Windows 10** virtual machine.
15. In the **Wireshark** window, click the **Stop** (■) button to stop capturing the packets.

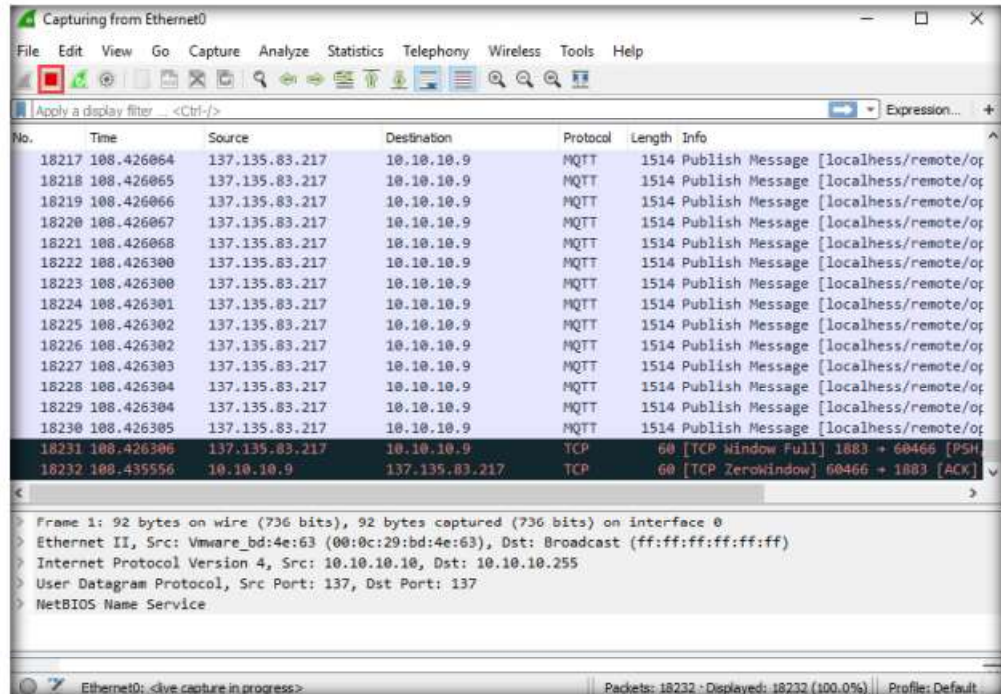


Figure 2.1.7: Stop packet capturing in Wireshark

16. Now, in the **Apply a display filter** field, type **mqtt** and press **Enter** to display only the MQTT protocol packets.

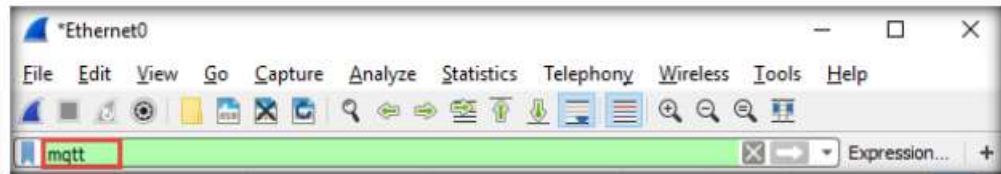


Figure 2.1.8: Filtering MQTT protocol traffic

17. You can observe packets such as **Connect Command**, **Connect Ack**, **Subscribe Request**, **Subscribe Ack**, and **Publish Message**, as shown in the screenshot.

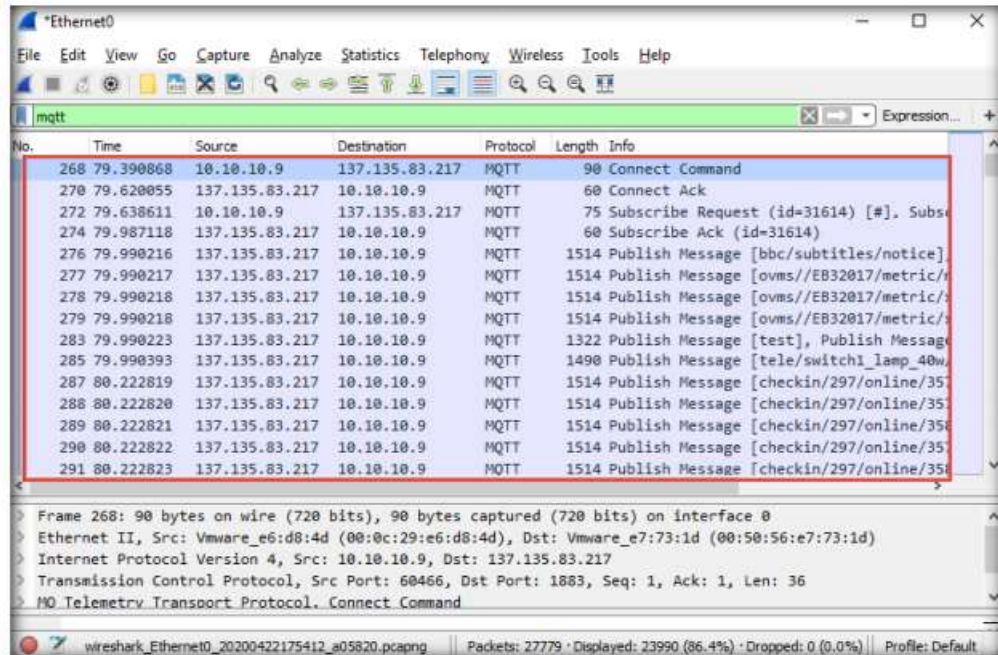


Figure 2.1.9: MQTT protocol packets

TASK 1.4

Analyze Connect Command Packet

18. Click on the **Connect Command** packet. In the lower section of the window, click to expand the **Transmission Control Protocol** and **MQ Telemetry Transport Protocol** nodes.
19. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Protocol Name**, **Version**, and **Client ID**.

Note: The MQTT protocol establishes a connection between clients through a broker. A CONNECT command is sent to initiate a connection from the client to the broker. After the connection is established, it remains active until a disconnect command is sent from the client.

Some of the headers of the CONNECT command are given below:

- **Header Flags:** Contains information regarding the MQTT control packet type
- **Connect Flags:** Contains parameters specifying the behavior of the MQTT connection
- **Clean Session:** Indicates whether the client wants to establish a persistent connection with the broker or not
- **Client ID:** Indicates a unique identifier for each MQTT client connecting to an MQTT broker

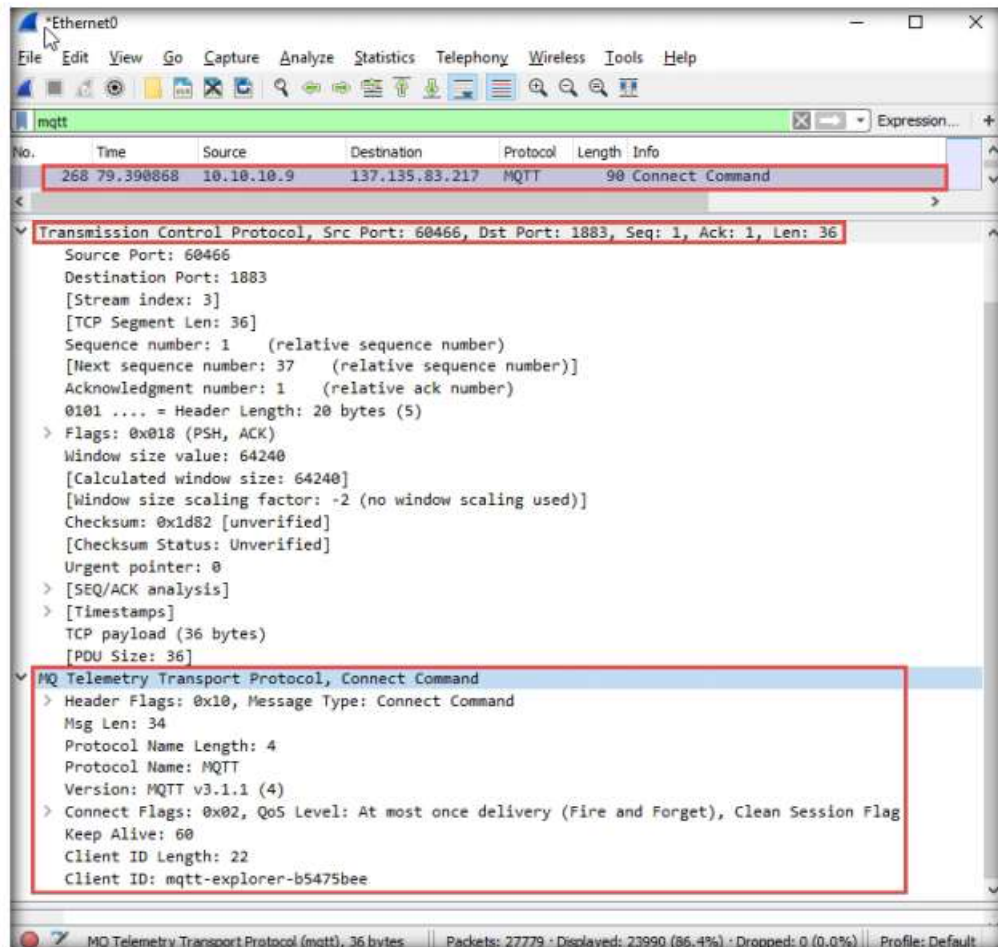


Figure 2.1.10: Connect Command packet

TASK 1.5

Analyze Connect Ack Packet

- Click on the **Connect Ack** packet. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
- Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Header Flag** and **Return Code**.

Note: The broker sends the Connect Ack packet on receiving a Connect command request from the client.

Some of the headers in the Connect Ack packet are given below:

- Header Flags:** Contains information regarding the MQTT control packet type
- Session Present:** Indicates the session between the broker and client; Bit 0 is the Connection Ack bit in the session present flag.
- Return Code:** The values and responses of the return code are summarized in the table below

Return Code	Return Code Response
0	Connection Accepted
1	Connection Refused, unacceptable protocol version
2	Connection Refused, identifier rejected
3	Connection Refused, server unavailable
4	Connection Refused, bad credentials
5	Connection Refused, not authorized

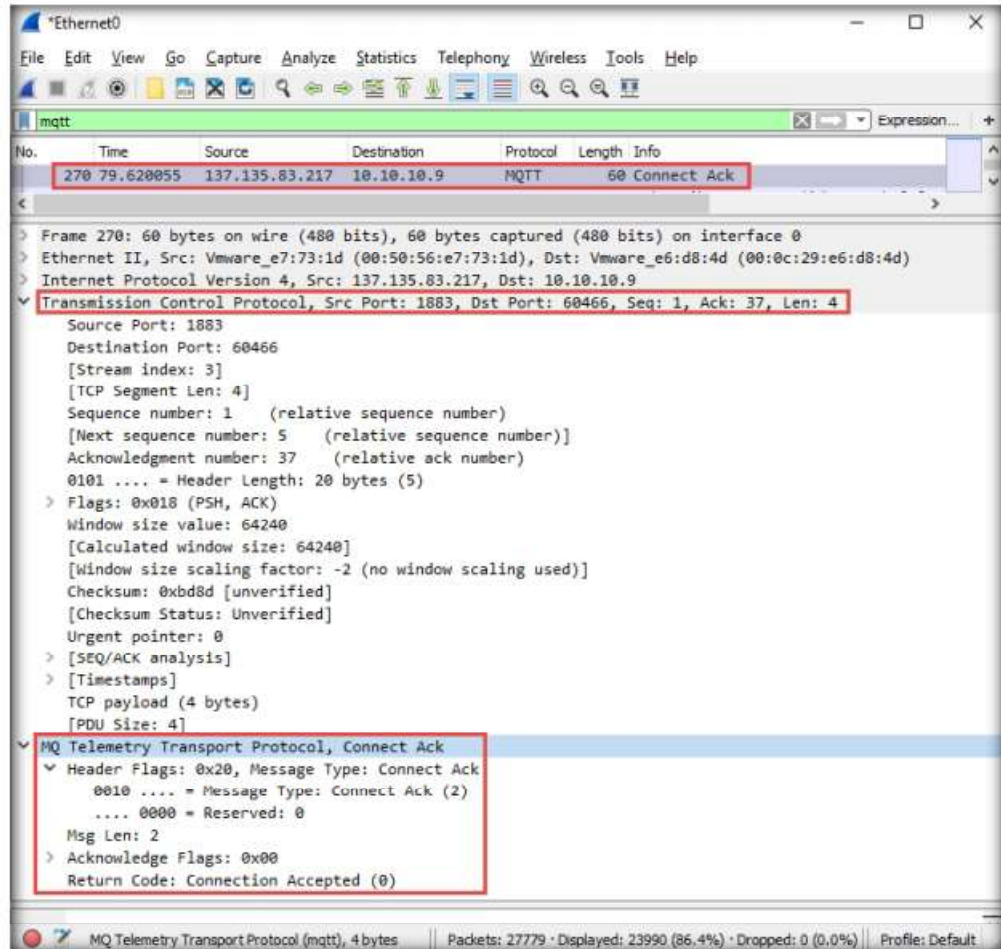


Figure 2.1.11: Connect Command packet

TASK 1.6

Analyze Subscribe Request Packet

22. Click on the **Subscribe Request** packet. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
23. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Message Identifier**, and **Topic Length**.

Note: To receive a relevant message, a client sends a SUBSCRIBE message to an MQTT broker.

Some of the headers in the Subscribe Request packet are given below:

- **Header Flags:** Contains information regarding the MQTT control packet type
- **Message Identifier:** Identifies a message in a message flow between a client and a broker
- **Topic and QoS Level:** A subscription is a pair of a topic filter and a QoS level; the topic defines a subject of interest on which the client would like to get messages
- **Payload:** Contains a list of subscriptions

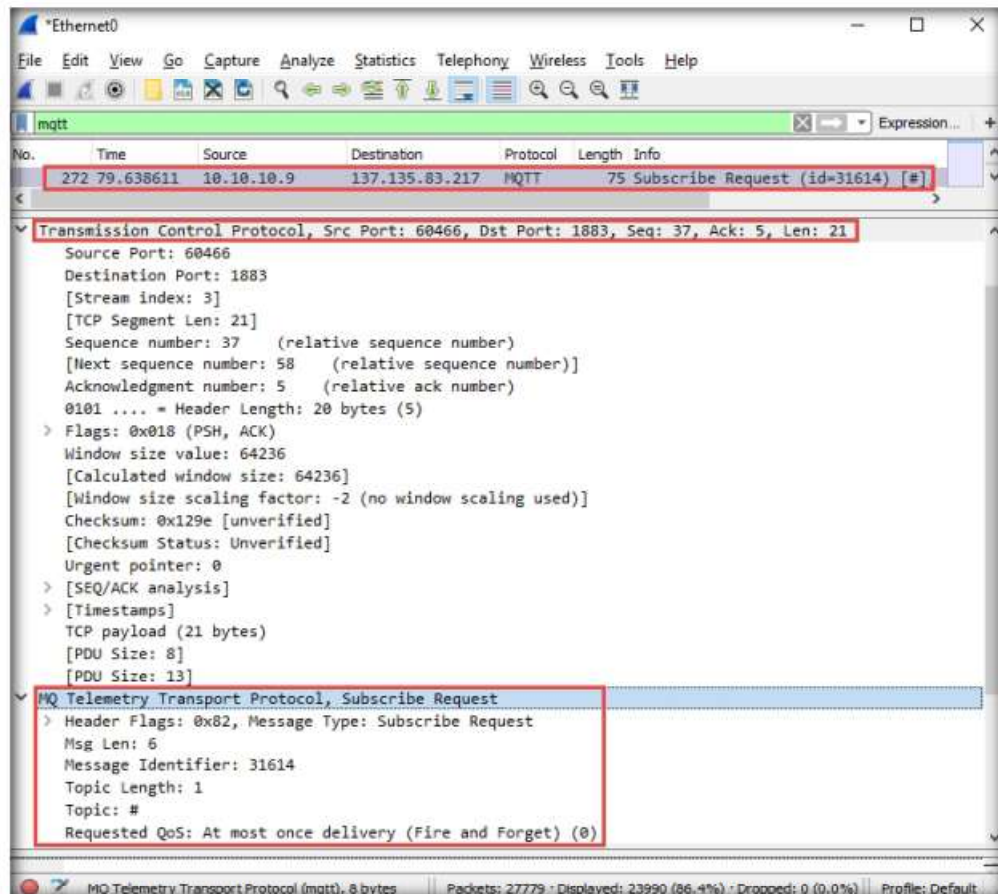


Figure 2.1.12: Subscribe Request packet

TASK 1.7

Analyze Subscribe Ack Packet

24. Click on the **Subscribe Ack** packet. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
25. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len** and **Message Identifier**.

Note: The MQTT broker confirms subscription by sending an acknowledgment back to the client using a SUBACK message.

Some of the headers in the Subscribe Acknowledgement packet are given below:

- **Header Flags:** Contains information regarding the MQTT control packet type
- **Message Identifier:** Identifies a message in a message flow between a client and a broker
- **Payload:** Contains a list of return codes
- **Return Code:** For each Topic/QoS pair received, a return is sent by the MQTT broker in the SUBSCRIBE message; the return code is in line with the QoS level in the case of a success

The values and responses of the return code are summarized in the table below:

Return Code	Return Code Response
0	Success - Maximum QoS 0
1	Success - Maximum QoS 1
2	Success - Maximum QoS 2
128	Failure

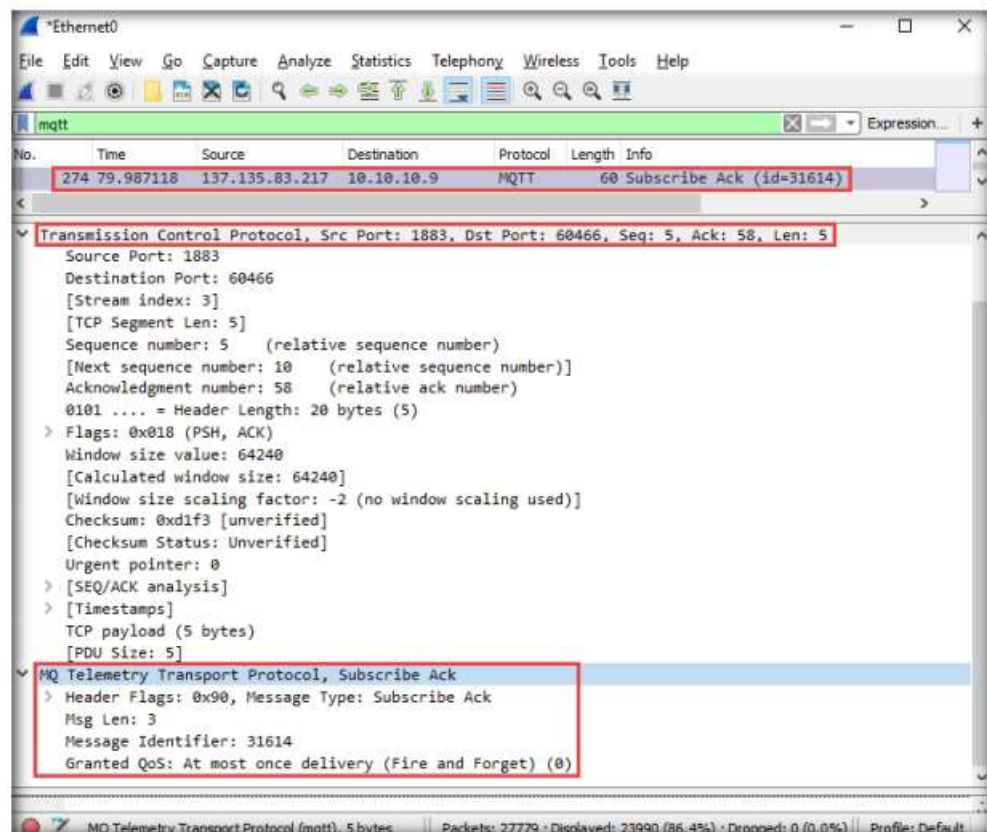


Figure 2.1.13: Subscribe Ack packet

TASK 1.8

Analyze Publish Message Packet

26. Click on any **Publish Message** packet. In the lower section of the window, click to expand the **Transmission Control Protocol, MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
27. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Topic Length**, **Topic**, and **Message**.

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

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

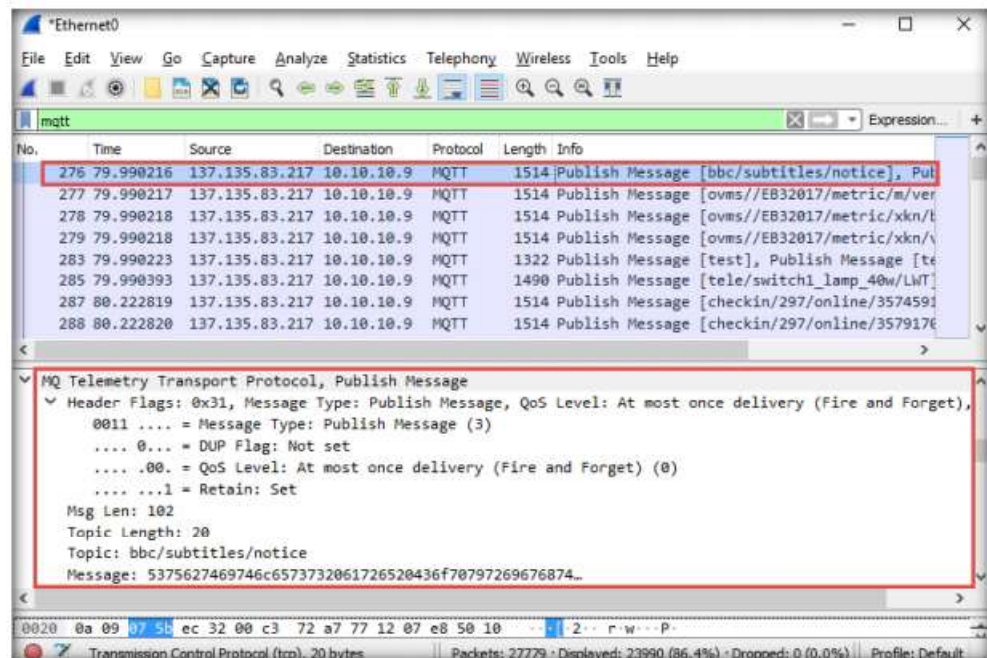


Figure 2.1.14: Publish Message packet

TASK 1.9

Analyze Publish Ack Packet

29. Click on any **Publish Ack** packet. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
30. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len** and **Message Identifier**.

Note: A Publish Ack (PUBACK) packet is the response to a Publish Message (PUBLISH) packet.

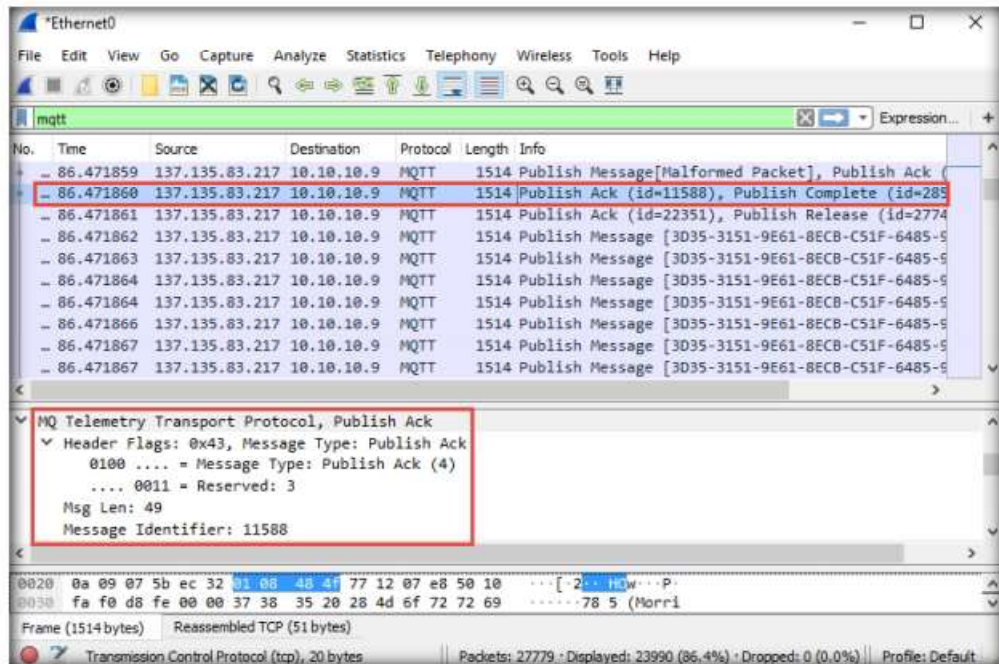


Figure 2.1.15: Publish Ack packet

TASK 1.10

Analyze Publish Release Packet

31. Click on any **Publish Release** packet. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
32. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len**, **Message Type**, **Message Identifier**.

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

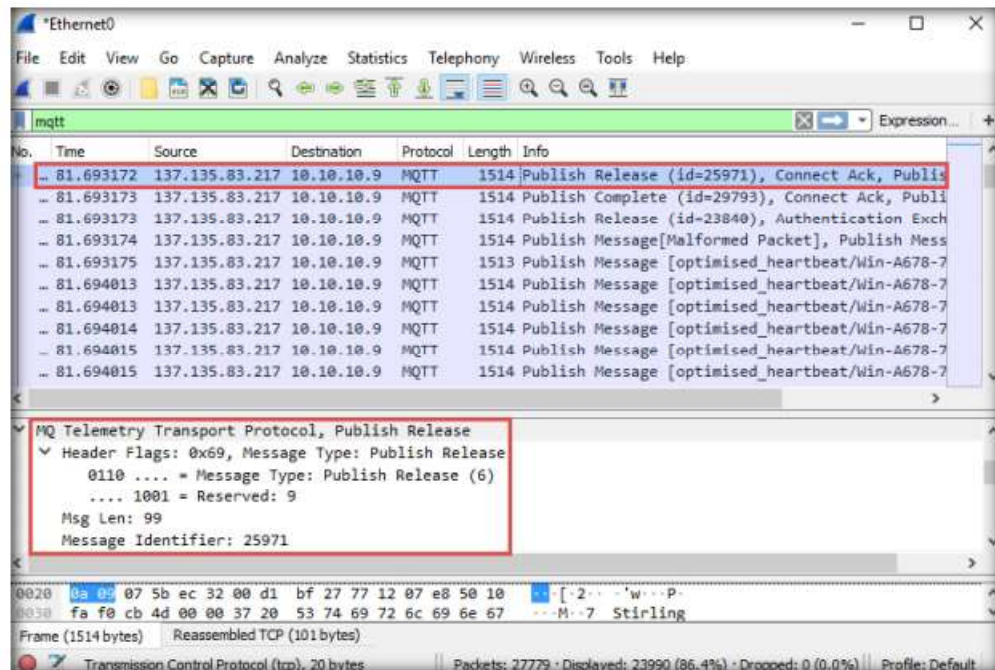


Figure 2.1.16: Publish Release packet

TASK 1.11

Analyze Publish Complete Packet

33. Now, scroll down, look for the **Publish Complete** packet, and click on it. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
34. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len** and **Message Identifier**.

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

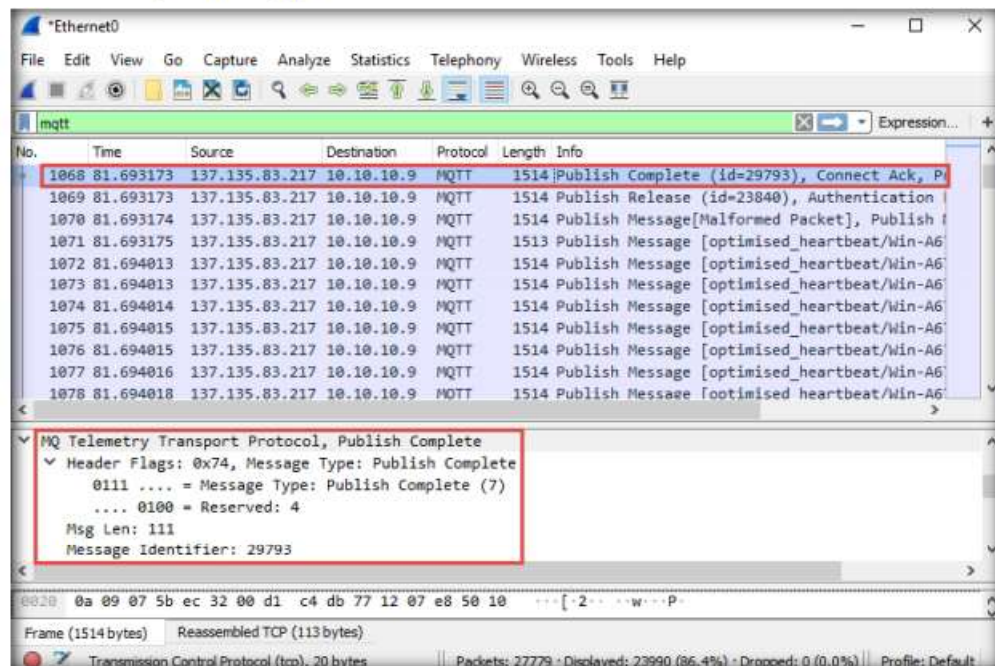


Figure 2.1.17: Publish Complete packet

TASK 1.1.2

Analyze
Disconnect Req
Packet

35. Scroll down, look for the **Disconnect Req** packet, and click on it. In the lower section of the window, click to expand the **Transmission Control Protocol**, **MQ Telemetry Transport Protocol**, and **Header Flags** nodes.
36. Under the **MQ Telemetry Transport Protocol** nodes, you can observe details such as **Msg Len** and **Message Identifier**.
37. A disconnect message is the final control packet sent by the client to the broker. This indicates a clean disconnection by the client.

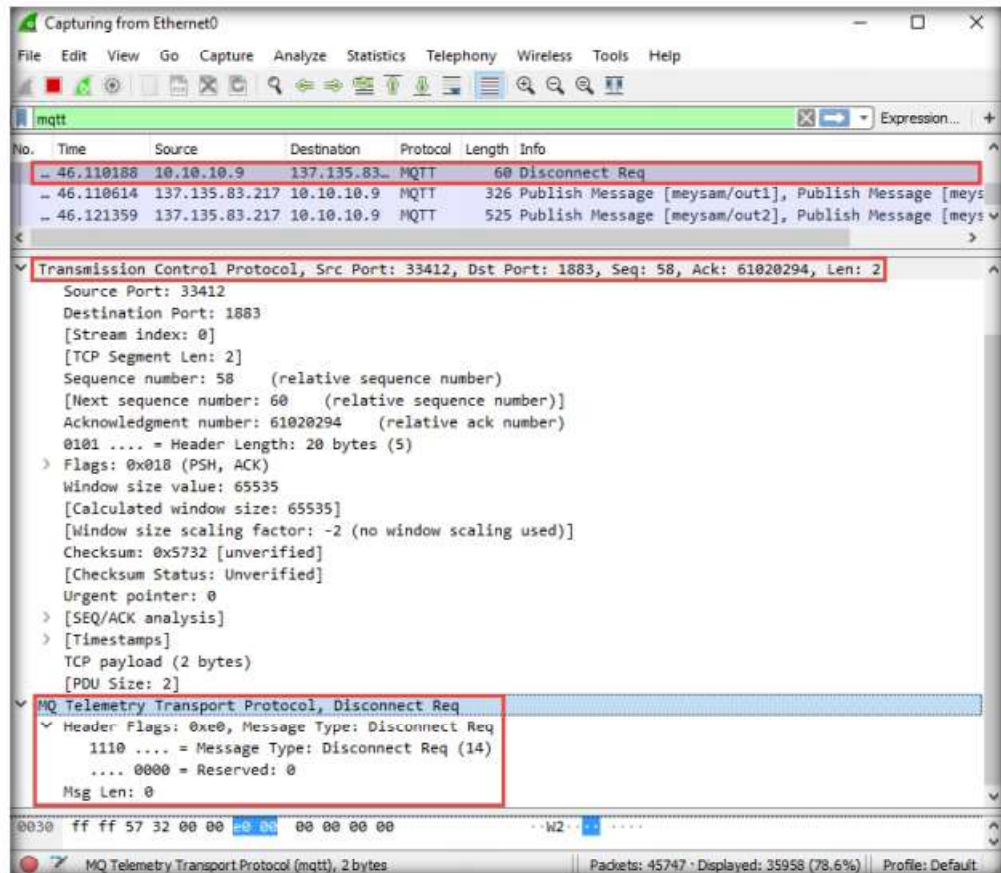


Figure 2.1.18: Disconnect Ask packet

38. 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.
39. Close all open windows and document all the acquired information.
40. Turn off the **Windows 10** and **Ubuntu** virtual machines.

Lab Analysis

Analyze and document all the results obtained in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE ANY
QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
Platform Supported	
<input checked="" type="checkbox"/> Classroom	<input checked="" type="checkbox"/> iLabs