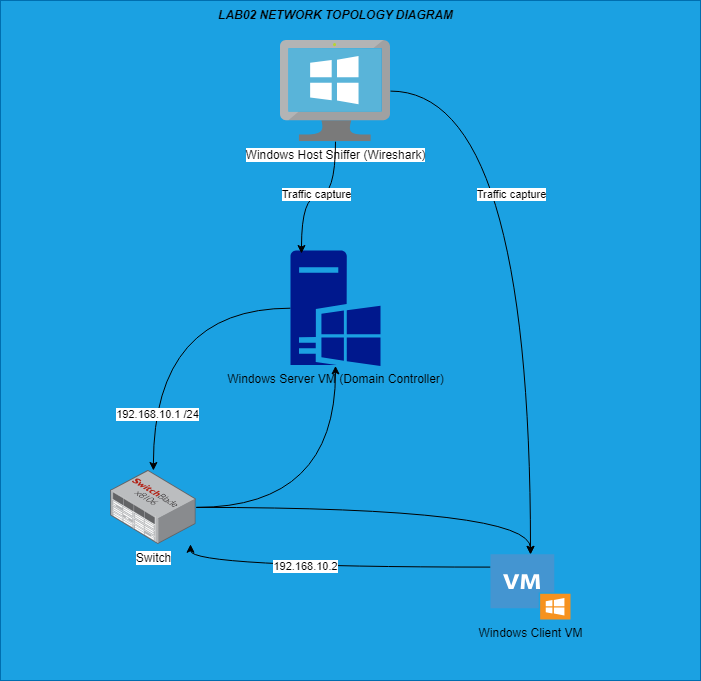
**Lab 02 – Analyzing Network Traffic using Packet Capture Software**

**Description**

This lab focuses on analyzing network traffic using packet capture software(Wireshark), to examine various protocols and communication patterns between a client and server. By capturing and analyzing DNS, ICMP, FTP, and TCP traffic, I aimed to understand how network communication occurs, identify potential issues, and ensure that the environment operates as expected. The results of the analysis provide valuable insights into network performance, security, and potential areas of vulnerability that could be leveraged by malicious actors.

**Topography**



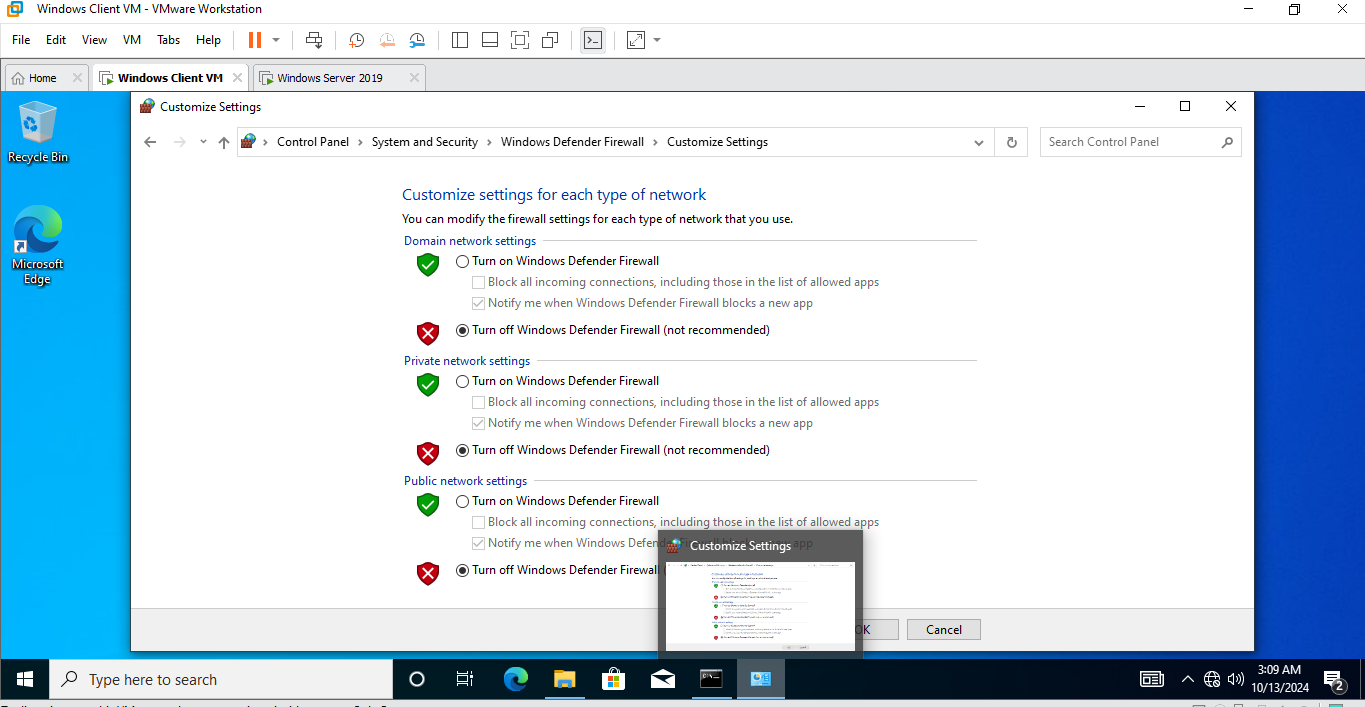
**Key Syntax**

|  |  |  |
| --- | --- | --- |
| **Task** | **Command/Syntax** | **Description** |
| **Check Network Connectivity** | Ping 192.168.10.1 | Checked connectivity between the client (192.168.10.2) and the server (192.168.10.1) using ICMP Echo requests. |
| **Perform DNS Lookup** | nslookup cyberlab.local | Checked if the client can resolve the domain (cyberlab.local) to an IP address. |
| **Ping Server from Client** | ping 192.168.10.1 | Verified server availability by sending ICMP Echo requests and receiving replies. |
| **Trace Path** | tracert 192.168.10.1 | Used the tracert command to trace the path between the Windows Client VM and the Windows Server VM. |
| **View ARP Cache (Client)** | arp -a | Viewed the ARP cache to confirm MAC-to-IP address mapping on the Windows Client VM. |
| **View ARP Cache (Server)** | arp -a | Checked the ARP cache on the Windows Server VM for MAC-to-IP address mapping. |
| **Remote Desktop Connection** | mstsc | Opened the Remote Desktop Connection from the client to the server (port 3389). |
| **Map Network Drive** | net use Z: \\192.168.10.1\share | Mapped a network drive from the client to the server for file sharing. |
| **FTP Commands** | ftp <IP Address>  USER <username>  PASS <password>  STOR <filename> | Connected to the FTP server, authenticated, and transferred files using the STOR command. |
| **ICMP Traffic Analysis** | ping 192.168.10.2 | Analyzed ICMP traffic with multiple Echo requests and replies between the client and server. |

**TASK ONE: VM and Wireshark Installation**

1. Disable Windows Firewall

I started by disabling the firewall for all network types—domain, private, and public—to ensure that there would be no interruptions during the network traffic analysis. I did this both on windows client and server.



A screenshot of a computer

Description automatically generated

**Verification**

**TASK TWO: Generate and Capture Network Traffic**

1. Ensure Network Connectivity

I confirmed network connectivity by successfully pinging the server from the client and vice versa, with no packet loss or latency issues.

A screenshot of a computer

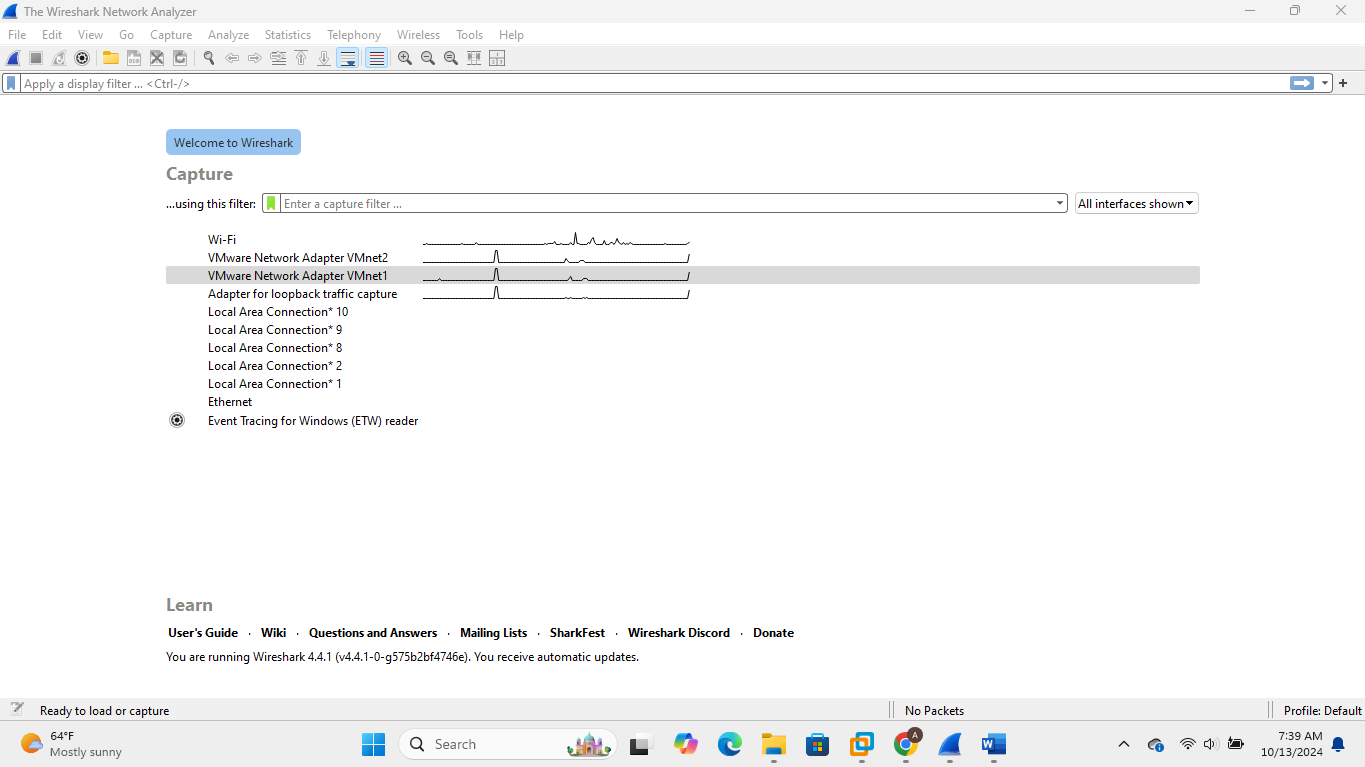
Description automatically generated

A computer screen with white text

Description automatically generated

1. Wireshark Installation

I installed Wireshark and ensured the necessary network adapters were ready for traffic capture.

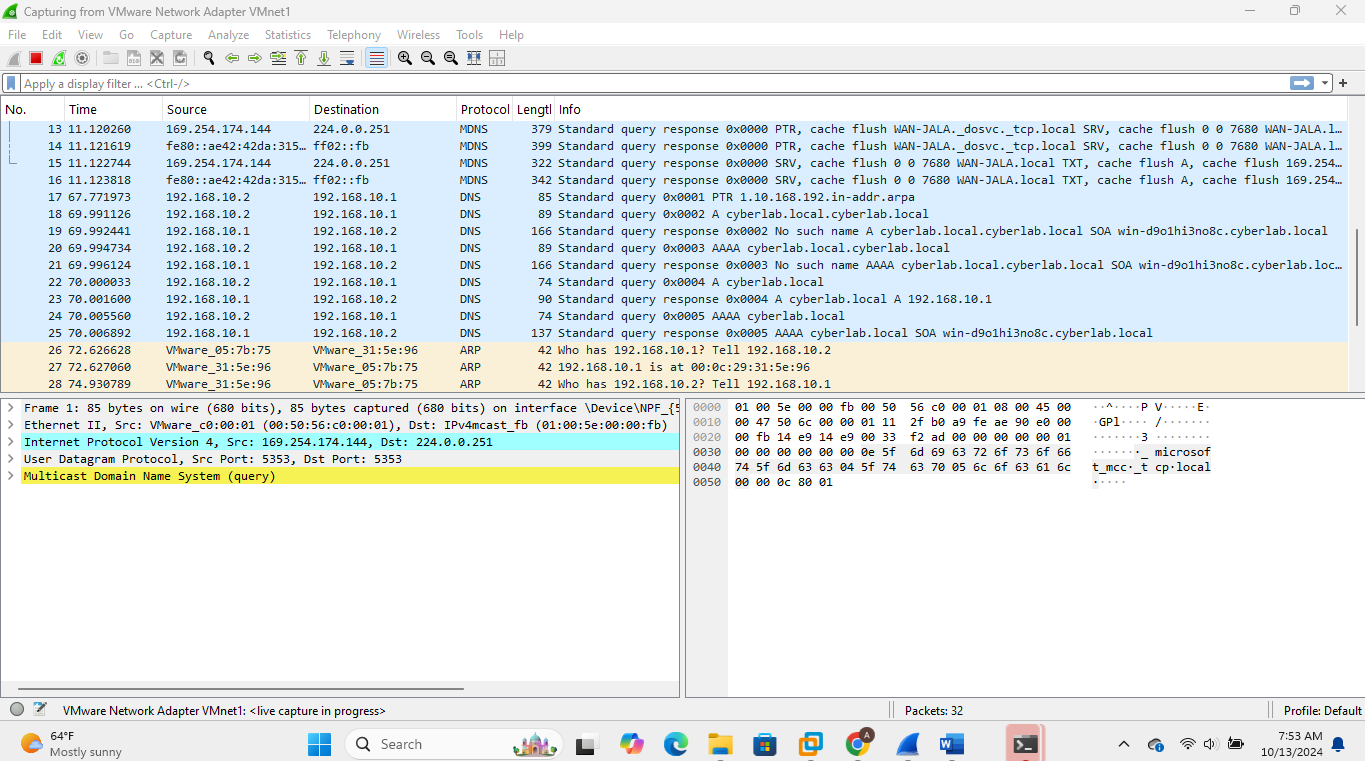


1. Perform DNS Lookup:

I start generating the network traffic by performing a DNS lookup from the Windows Client VM to ensure that the client could resolve the server’s domain name to an IP address.

A computer screen with a black background

Description automatically generated



1. Ping the Windows Server from the Client:

I verified the server’s availability by sending a ping request from the Windows Client VM to the Windows Server VM, receiving successful replies.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Trace the Path

I used the tracert command from the Windows Client VM to trace the route to the Windows Server VM, confirming the path between both machines.

A computer screen with text on it

Description automatically generated

A screenshot of a computer

Description automatically generated

1. View the ARP cache on the Windows Client VM

I viewed the ARP cache on the Windows Client VM to confirm that the MAC address of the Windows Server VM was mapped to its IP address.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. View the ARP cache on the Windows Server VM

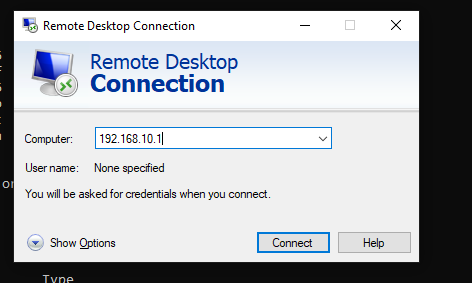
I accessed the ARP cache on the Windows Server VM to verify that the MAC address of the Windows Client VM was correctly associated with its IP address.

A computer screen with text on it

Description automatically generated

1. Remote Desktop

I successfully opened a Remote Desktop Connection from the Windows Client VM to the Windows Server VM, confirming that remote access was functional.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

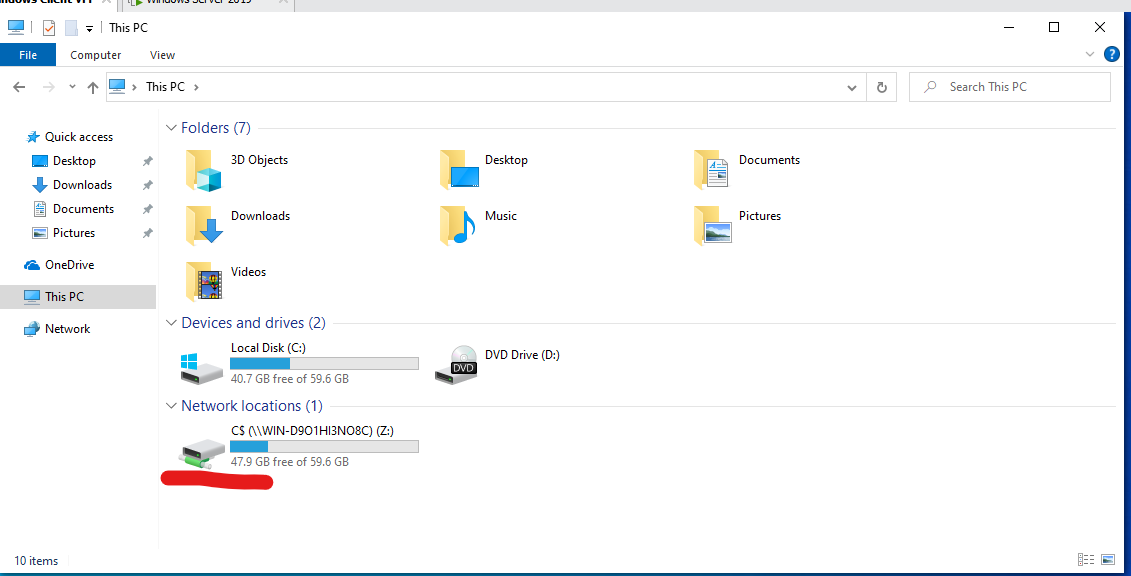
Description automatically generated

1. Map Network Drive

I mapped a network drive from the Windows Client VM to the Windows Server VM, enabling easy file access and sharing between the two systems.

A screen shot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

1. FTP

I then initiated an FTP connection from the Windows Client VM to the Windows Server VM and successfully uploaded a file to generate more traffic.

A screenshot of a computer

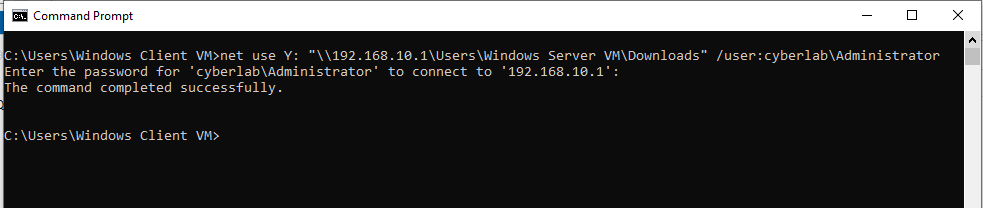
Description automatically generated

A screenshot of a computer

Description automatically generated

1. File Sharing via Network Drive Mapping

Finally, I transferred a file through the mapped network drive from the Windows Client VM to the Windows Server VM to confirm successful file sharing and generate additional traffic.

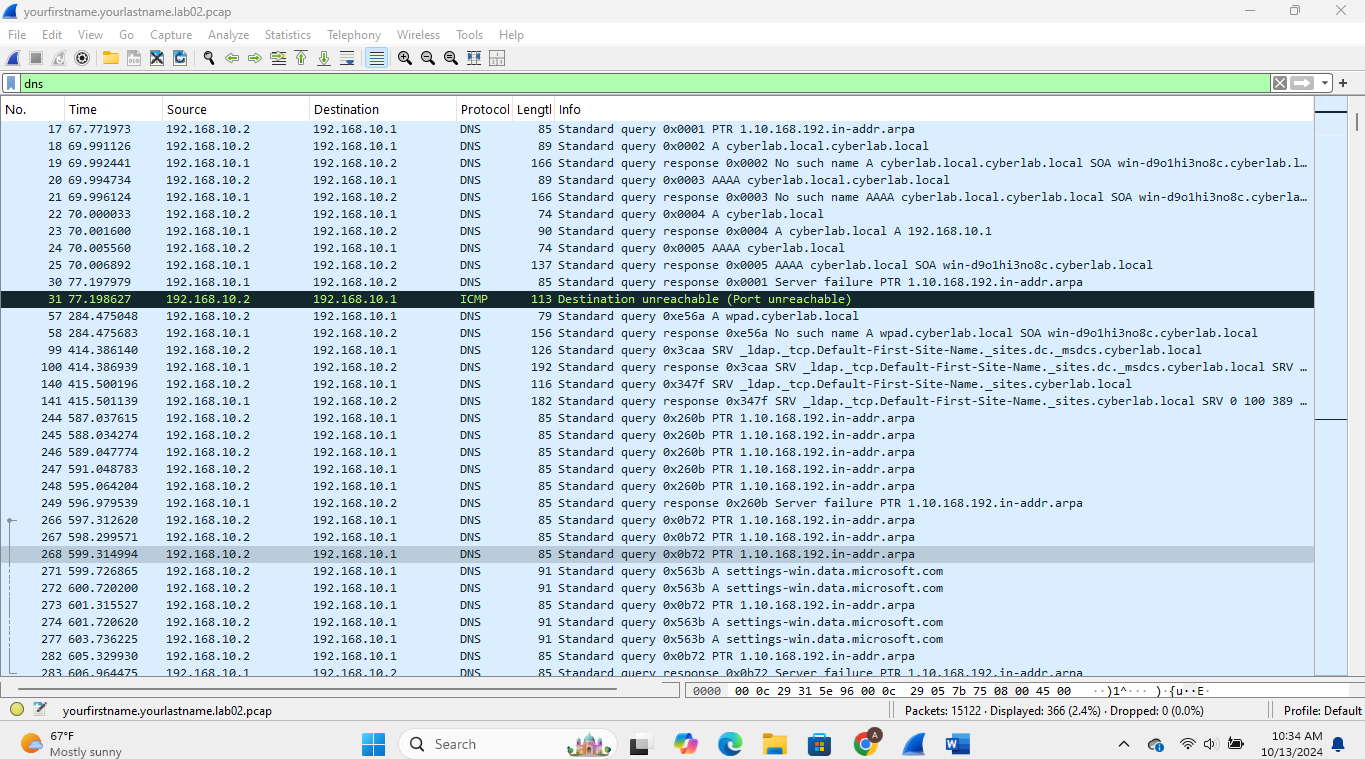


A screenshot of a computer

Description automatically generated

**TASK THREE: Analyze Network Traffic**

**DNS Traffic Analysis Summary**



In analyzing the DNS traffic between the client and server, I observed that while the server successfully resolved internal domain queries (like cyberlab.local), external queries, such as those for [www.bing.com](http://www.bing.com), consistently failed, likely due to misconfigurations or lack of internet access. The DNS protocol functioned as expected, with the client sending requests and the server replying, but the repeated failures for external queries were surprising. This tool provided valuable insight into potential network issues and could also be leveraged by bad actors to identify weaknesses in DNS configurations for exploitation.

**ICMP Traffic Analysis**

A screenshot of a computer

Description automatically generated

I analyzed the ICMP traffic between the client and server by sending multiple ping requests from the client, which the server successfully responded to with echo replies. However, several "Destination unreachable (Port unreachable)" responses were observed, indicating that some requests were sent to closed ports on the server. This was somewhat unexpected but highlights the importance of properly configuring open ports for communication. The intelligence gained from this analysis could be valuable for a bad actor to identify potential vulnerabilities, such as closed ports, to exploit for further probing or attacks.

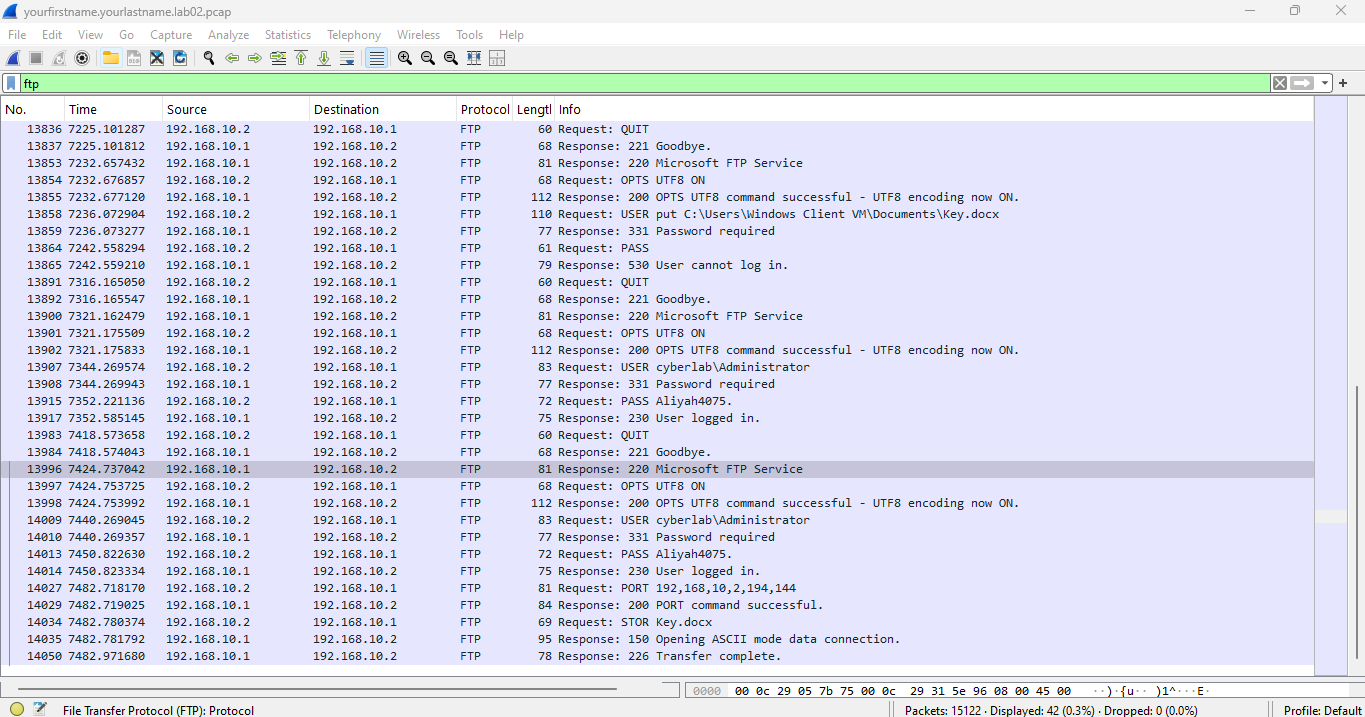
**TCP Traffic Analysis**

A screenshot of a computer

Description automatically generated

I analyzed the TCP traffic between the client and server on port 3389 (RDP). Initially, the client sent SYN packets to start the connection, but the server responded with RST, indicating the port was closed. Later, the connection was successfully established, and encrypted data was exchanged over TLSv1.2. Some retransmissions and duplicate acknowledgments occurred, likely due to network issues, but overall, the RDP session progressed as expected. This information is useful for identifying potential connectivity problems, and a bad actor could use it to detect open or closed ports to exploit vulnerabilities.

**FTP Traffic Analysis**



During my analysis of the FTP traffic between the client (192.168.10.2) and the server (192.168.10.1), I observed the client making multiple FTP requests to the server, starting with a connection initialization. The server responded with a "220 Microsoft FTP Service" message, indicating that the FTP service was available. The client then successfully requested the "OPTS UTF8 ON" option, which allowed UTF8 encoding for future commands.

I noticed several login attempts, such as the command "USER put C:\Users\Windows Client VM\Documents\Key.docx" followed by the server response "331 Password required." The client then attempted to log in with the password "Aliyah4075", but the login failed multiple times, returning "530 User cannot log in" as a response. After several retries, the client successfully logged in with the credentials "cyberlab\Administrator" and the password "Aliyah4075," receiving the response "230 User logged in."

Following a successful login, the client initiated a file transfer using the "STOR Key.docx" command. The server acknowledged with "150 Opening ASCII mode data connection" and later confirmed the file transfer was successful with the "226 Transfer complete" message. This demonstrated that the file upload was completed without any issues.

**SMB and Host Announcements Analysis**

A screenshot of a computer

Description automatically generated

During my analysis of the traffic logs, I observed frequent host announcements from various devices on the network, as well as SMB (Server Message Block) protocol negotiation requests. The client devices, including DESKTOP-G44LU9T, WIN-D9O1HI3NO8C, and WAN-JALA, regularly broadcasted host announcements across the network. These broadcasts occurred at intervals, as seen in packet numbers like 211, 236, 595, and many others. The announcements indicated the presence of services like Workstation, Server, NT Workstation, Domain Controller, Time Source, and DFS Server, depending on the host.

A notable observation was the host WAN-JALA using an IP address from the 169.254.x.x range, which suggests that it was operating under an APIPA (Automatic Private IP Addressing) configuration, possibly due to a DHCP failure. It repeatedly broadcasted its availability, yet its lack of a properly assigned IP indicates potential network issues or misconfigurations.

There were multiple SMB Negotiate Protocol Request messages initiated by 192.168.10.2 to 192.168.10.1. These negotiation requests indicate attempts by the client to establish a session with the server. This negotiation is a critical part of the SMB protocol, which is often used for file and printer sharing in networked environments. These requests were observed consistently over time, implying ongoing communication or attempts to establish or re-establish sessions with the server.

The pattern of these SMB negotiations, paired with the host announcements, points to a healthy, yet actively monitored, network environment where devices are broadcasting their services, and SMB protocol traffic is continuously occurring to maintain file-sharing services.

**ARP Traffic Analysis**

A screenshot of a computer

Description automatically generated

The ARP traffic between the client (192.168.10.2) and server (192.168.10.1) reveals consistent address resolution requests and responses, indicating normal network behavior. Both systems regularly query each other's MAC addresses, responding accurately with their VMware MAC addresses, 00:0c:29:05:7b:75 and 00:0c:29:31:5e:96, respectively. This pattern confirms that the network connection is stable, with no indication of address resolution issues or ARP cache problems.

**Conclusion**

In this lab, I analyzed various network protocols, including DNS, ICMP, TCP, FTP, SMB, and ARP, using Wireshark to understand the communication patterns between the client and server. The analysis revealed normal internal traffic behavior, though external queries and closed ports indicated potential network misconfigurations. The insights gained can help identify and address performance issues, enhance security, and reduce vulnerabilities that could be exploited by malicious actors.

**References**

* Combs, G. (2023). *Wireshark User's Guide*. Riverbed Technology. https://www.wireshark.org/docs/wsug\_html\_chunked/
* Easttom, C. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
* Stallings, W. (2020). *Network Security Essentials: Applications and Standards* (6th ed.). Pearson.
* Forouzan, B. A. (2017). *Data Communications and Networking* (5th ed.). McGraw-Hill Education.
* Kurose, J. F., & Ross, K. W. (2021). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.