Explanation of the Subject

Chapter II: Introduction

The subject introduces the OSI model, which is a conceptual framework used to understand network interactions in seven layers. Each layer has its own vulnerabilities and risks. The focus here is on the network layer, where routers (default gateways) direct traffic. If a network node can impersonate the gateway, it can control, intercept, modify, or block traffic. This is known as ARP spoofing, which can be used maliciously or legitimately (e.g., redirecting users to a registration page in public networks).

Chapter III: Mandatory Part

The project involves creating a program called "inquisitor" that performs ARP poisoning and monitors FTP traffic. Key requirements include:

- **Platform**: Linux
- **Parameters**: IP and MAC addresses for source and target
- **Protocol**: IPv4 only
- **Error Handling**: Must handle all input errors and not crash unexpectedly
- **Testing**: Must include tests using the FTP protocol

The program must:

- Perform ARP poisoning in both directions (full duplex)
- Restore ARP tables when the attack is stopped (e.g., via CTRL+C)
- Display filenames exchanged between an FTP client and server in real-time

Chapter IV: Bonus Part

An optional enhancement includes a verbose mode (-v) that shows all FTP traffic, including login details. This bonus will only be assessed if the mandatory part is perfect.

Documentation and Resources

To successfully complete this project, you should familiarize yourself with the following topics and resources:

ARP Spoofing and Poisoning:

- ARP Spoofing - Wikipedia
- ARP Poisoning - OWASP

OSI Model:

- OSI Model - Wikipedia
- Understanding the OSI Model - Cisco

Libpcap Library:

- Libpcap Documentation
- Libpcap Tutorial

Docker and Containers:

- Docker Documentation
- Dockerfile Reference
- Docker Compose Documentation

Makefile:

- GNU Make Manual
- Makefile Tutorial

FTP Protocol:

- FTP - Wikipedia
- RFC 959 - FTP

Programming Languages:

Python:

- Python Libpcap Bindings
- Scapy Documentation

C/C++:

- Libpcap Programming in C

Steps to Implement the Project

Setup Environment:

- Create a Dockerfile and docker-compose.yaml to set up the environment.
- Write a Makefile to automate the setup and execution.

Develop the Program:

- Implement ARP poisoning using libpcap.
- Ensure the program handles errors gracefully and restores ARP tables on exit.
- Capture and display FTP filenames in real-time.

Testing:

- Write test cases to validate the program using FTP connections.
- Ensure the program works as expected in various scenarios.

Bonus Implementation:

- Add a verbose mode to capture and display all FTP traffic, including login details.