ARP Poisoning and Man-in-the-Middle (MITM) Attack Simulation

Introduction

This project focuses on demonstrating an **Address Resolution Protocol (ARP) poisoning** attack, a common technique used in **Man-in-the-Middle (MITM)** scenarios. By exploiting vulnerabilities in the ARP protocol, this attack allows an adversary to intercept, inspect, modify, or block communication between devices on the same local network.

The goal of this project is to showcase the ease with which network communication can be compromised using ARP poisoning and to emphasize the importance of implementing countermeasures such as ARP spoofing detection and encrypted communication protocols.

Background Information

What is ARP?

Address Resolution Protocol (ARP) is a protocol used to map an IP address to a MAC address within a local area network (LAN). Devices in a LAN use ARP to resolve the hardware address of a destination device before sending data.

- ARP works at **Layer 2** (Data Link Layer) of the OSI model.
- It sends a broadcast request to all devices in the network, asking "Who has this IP address?" The device with the matching IP responds with its MAC address.

What is a Man-in-the-Middle (MITM) Attack?

A **Man-in-the-Middle (MITM)** attack occurs when an attacker intercepts and manipulates communication between two parties without their knowledge. In ARP poisoning:

1. The attacker poisons the ARP cache of a target device, associating the attacker's MAC address with the IP address of a legitimate device.
2. Traffic meant for the legitimate device is rerouted through the attacker’s machine, enabling data inspection, modification, or blocking.

Project Overview

This project implements an ARP poisoning attack to demonstrate:

1. The fundamental vulnerabilities of the ARP protocol.
2. The steps required to execute an ARP poisoning attack.
3. Real-world implications of such attacks on data privacy and security.

Implementation Details

1. Setting Up the Environment

- **Network Configuration**:
  - Two virtual machines (VMs) are created on a **NAT-configured network**.
  - The host computer acts as the default gateway.
- **Software Tools**:
  - The attack is implemented using a custom C program.
  - Tools like Wireshark are used to verify the attack.

---

2. Attack Workflow

a. Initialize ARP and Ethernet Headers

The program begins by defining and configuring the **Ethernet** and **ARP** headers:

- **Ethernet Header**:
  - Source MAC: Attacker’s MAC address.
  - Destination MAC: Target’s MAC address or broadcast (`FF:FF:FF:FF:FF:FF`).
- **ARP Header**:
  - Set operation to **ARP Reply (ARPOP_REPLY)**.
  - Assign the attacker’s MAC address to the spoofed IP address.

b. Create and Configure a Raw Socket

A raw socket is created to send low-level network packets. This socket operates at Layer 2, bypassing the typical restrictions of Layer 3.

c. Send Forged ARP Packets

The program repeatedly sends forged ARP packets to the target device. Each packet updates the target's ARP cache, associating the attacker's MAC address with the legitimate device's IP address.

d. Intercept and Manipulate Traffic

Once the ARP poisoning is successful, traffic intended for the legitimate device is redirected to the attacker. Using tools like Wireshark, the attacker can capture and analyze the intercepted data.

---

3. Verification and Results

a. ARP Cache Inspection

On the target device, running the command `arp -a` reveals that the spoofed IP address is now associated with the attacker's MAC address, confirming that the ARP cache has been poisoned.

b. Packet Analysis

Using Wireshark on the attacker's machine shows incoming traffic originally destined for the legitimate device. This confirms that the attack successfully reroutes traffic through the attacker's machine.

Results

The ARP poisoning attack successfully demonstrates the vulnerability of the ARP protocol and the feasibility of a MITM attack:

- Targeted devices accept the poisoned ARP responses without verification.
- Traffic between devices can be intercepted and manipulated by the attacker.

Implications

1. **Security Risks**:
    - Sensitive information (e.g., login credentials, personal data) can be intercepted.
    - Communication integrity is compromised, allowing attackers to modify or inject data.
2. **Preventive Measures**:

- **Static ARP Tables**: Assign static mappings of IP and MAC addresses.
- **Encryption**: Use secure protocols like HTTPS or VPNs to prevent eavesdropping.
- **ARP Spoofing Detection**: Tools like ARPwatch or IDS systems can detect anomalies