

## 26. ARP

### 26. Wired Local Networks: ARP

#### 26.1. Cheat sheet

- Layer: Link (2)
- Purpose: Translate IP addresses to MAC addresses
- Vulnerability: On-path attackers can see requests and send spoofed malicious responses
- Defense: Switches, arpswatch

#### 26.2. Networking background: Ethernet

Recall that on a LAN (local-area network), all machines are connected to all other machines. Ethernet is one particular LAN implementation that uses wires to connect all machines.

Ethernet started as a broadcast-only network. Each node on the network could see messages sent by all other nodes, either by being on a common wire or a network **hub**, a simple repeater that took every packet it received and rebroadcast it to all the outputs. A receiver is simply supposed to ignore all packets not sent to either the receiver's MAC or the broadcast address. But this is only enforced in software, and most Ethernet devices can enter **promiscuous mode**, where it will receive all packets. This is also called **sniffing packets**.

For versions of Ethernet that are inherently broadcast, such as a hub, an adversary in the local network can see all network traffic and can also introduce any traffic they desire by simply sending packets with a spoofed MAC address. Sanity check: what type of adversary does this make someone on the same LAN network as a victim?<sup>1</sup>

#### 26.3. Protocol: ARP

**ARP**, the **Address Resolution Protocol**, translates Layer 3 IP addresses into Layer 2 MAC addresses.

Say Alice wants to send a message to Bob, and Alice knows that Bob's IP address is 1.1.1.1. The ARP protocol would follow three steps:

1. Alice would broadcast to everyone else on the LAN: "What is the MAC address of 1.1.1.1?"
2. Bob responds by sending a message only to Alice: "My IP is 1.1.1.1 and my MAC address is `ca:fe:f0:0d:be:ef`." Everyone else does nothing.
3. Alice caches the IP address to MAC address mapping for Bob.

If Bob is outside of the LAN, then the router would respond in step 2 with its MAC address.

Any received ARP replies are always cached, even if no broadcast request (step 1) was ever made.

---

<sup>1</sup>A: On-path

## 26.4. Attack: ARP Spoofing

Because there is no way to verify that the reply in step 2 is actually from Bob, it is easy to attack this protocol. If Mallory is able to create a spoofed reply and send it to Alice before Bob can send his legitimate reply, then she can convince Alice that a different MAC address (such as Mallory's) corresponds to Bob's IP address. Now, when Alice wants to send a local message to Bob, she will use the malicious cached IP address to MAC address mapping, which might map Bob's IP address to Mallory's MAC address. This will cause messages intended for Bob to be sent to Mallory. Sanity check: what type of adversary is Mallory after she executes an ARP spoof attack?<sup>2</sup>

ARP spoofing is our first example of a race condition, where the attacker's response must arrive faster than the legitimate response to fool the victim. This is a common pattern for on-path attackers, who cannot block the legitimate response and thus must race to send their response first.

## 26.5. Defenses: Switches

A simple defense against ARP spoofing is to use a tool like `arpwatch`, which tracks the IP address to MAC address pairings across the LAN and makes sure nothing suspicious happens.

Modern wired Ethernet networks defend against ARP spoofing by using **switches** rather than hubs. Switches have a MAC cache, which keeps track of the IP address to MAC address pairings. If the packet's IP address has a known MAC in the cache, the switch just sends it to the MAC. Otherwise, it broadcasts the packet to everyone. Smarter switches can filter requests so that not every request is broadcast to everyone.

Higher-quality switches include **VLANs** (Virtual Local Area Networks), which implement isolation by breaking the network into separate virtual networks.

---

<sup>2</sup>A: Man-in-the-middle. She can receive messages from Alice, modify them, then send them to Bob.