Anatomy of an ARP Poisoning Attack

by Corey Nachreiner, WatchGuard Network Security Analyst

Hackers lie. Skillful hackers lie well. And well-rounded hackers can lie both to people and to machines.

Lying to people, known as "social engineering," involves tactics (detailed at length by convicted hacker Kevin Mitnick) such as posing as a company's employee so the company's real employees will blab secrets freely. Lying to machines involves lots of different techniques, and a commonly used one -- ARP Cache Poisoning -- is the focus of this article. ARP poisoning enables local hackers to cause general networking mayhem. Because it's mostly "incurable," every administrator should be aware of how this attack works.

ARP Refresher

In Foundations: What Are NIC, MAC, and ARP?, we explained that Address Resolution Protocol (ARP) is how network devices associate MAC addresses with IP Addresses so that devices on the local network can find each other. ARP is basically a form of networking roll call.

ARP, a very simple protocol, consists of merely four basic message types:

1. **An ARP Request**. Computer A asks the network, "Who has this IP address?"
2. **An ARP Reply**. Computer B tells Computer A, "I have that IP. My MAC address is [whatever it is]."
3. **A Reverse ARP Request (RARP)**. Same concept as ARP Request, but Computer A asks, "Who has this MAC address?"
4. **A RARP Reply**. Computer B tells Computer A, "I have that MAC. My IP address is [whatever it is]"

All network devices have an *ARP table*, a short-term memory of all the IP addresses and MAC addresses the device has already matched together. The ARP table ensures that the device doesn't have to repeat ARP Requests for devices it has already communicated with.

Here's an example of a normal ARP communication. Jessica, the receptionist, tells Word to print the latest company contact list. This is her first print job today. Her computer (IP address 192.168.0.16) wants to send the print job to the office's HP LaserJet printer (IP address 192.168.0.45). So Jessica's computer broadcasts an ARP Request to the entire local network asking, "Who has the IP address, 192.168.0.45?" as seen in Diagram 1.

All the devices on the network ignore this ARP Request, except for the HP LaserJet printer. The printer recognizes its own IP in the request and sends an ARP Reply: "Hey, my IP address is 192.168.0.45. Here is my MAC address: 00:90:7F:12:DE:7F," as in Diagram 2.

Now Jessica's computer knows the printer's MAC address. It sends the print job to the correct device, and it also associates the printer's MAC address of 00:90:7F:12:DE:7F with the printer's IP address of 192.168.0.45 in its ARP table.

Hey ARP, Did You Know Gullible Is Not in the Dictionary?

The founders of networking probably simplified the communication process for ARP so that it would function efficiently. Unfortunately, this simplicity also leads to major insecurity. Know why my short description of ARP doesn't mention any sort of authentication method? Because in ARP, there is none.

ARP is very trusting, as in, gullible. When a networked device sends an ARP request, it simply trusts that when the ARP reply comes in, it really does come from the correct device. ARP provides no way to verify that the responding device is really who it says it is. In fact, many operating systems implement ARP so trustingly that devices that have not made an ARP request still accept ARP replies from other devices.

OK, so think like a malicious hacker. You just learned that the ARP protocol has no way of verifying ARP replies. You've learned many devices accept ARP replies before even requesting them. Hmmm. Well, why don't I craft a perfectly valid, yet malicious, ARP reply containing any arbitrary IP and MAC address I choose? Since my victim's computer will blindly accept the ARP entry into its ARP table, I can force my victim's gullible computer into thinking any IP is related to any MAC address I want. Better yet, I can *broadcast* my faked ARP reply to my victim's entire network and fool *all* his computers. Muahahaha*haa!*

Back to reality. Now you probably understand why this common technique is called ARP Cache Poisoning (or just ARP Poisoning): the attacker lies to a device on your network, corrupting or "poisoning" its understanding of where other devices are. This frighteningly simple procedure enables the hacker to cause a variety of networking woes, described next.

All Your ARP Are Belong To Us!

The ability to associate any IP address with any MAC address provides hackers with many attack vectors, including Denial of Service, Man in the Middle, and MAC Flooding.

Denial of Service

A hacker can easily associate an operationally significant IP address to a false MAC address. For instance, a hacker can send an ARP reply associating your network router's IP address with a MAC address that doesn't exist. Your computers believe they know where your default gateway is, but in reality they're sending any packet whose destination is not on the local segment, into the Great Bit Bucket in the Sky. In one move, the hacker has cut off your network from the Internet.

Man in the Middle

A hacker can exploit ARP Cache Poisoning to intercept network traffic between two devices in your network. For instance, let's say the hacker wants to see all the traffic between your computer, 192.168.0.12, and your Internet router, 192.168.0.1. The hacker begins by sending a malicious ARP "reply" (for which there was no previous request) to your router, associating his computer's MAC address with 192.168.0.12 (see

Diagram 3).

Now your router thinks the *hacker's* computer is *your* computer.

Next, the hacker sends a malicious ARP reply to *your* computer, associating his MAC Address with 192.168.0.1 (see Diagram 4).

Now your machine thinks the hacker's *computer* is your *router*.

Finally, the hacker turns on an operating system feature called *IP forwarding*. This feature enables the hacker's machine to forward any network traffic it receives from your computer to the router (shown in Diagram 5).

Now, whenever you try to go to the Internet, your computer sends the network traffic to the hacker's machine, which it then forwards to the real router. Since the hacker is still forwarding your traffic to the Internet router, you remain unaware that he is intercepting all your network traffic and perhaps also sniffing your clear text passwords or hijacking your secured Internet sessions.

MAC Flooding

MAC Flooding is an ARP Cache Poisoning technique aimed at network switches. (If you need a reminder about the difference between a hub and a switch, see this sidebar.) When certain switches are overloaded they often drop into a "hub" mode. In "hub" mode, the switch is too busy to enforce its port security features and just broadcasts all network traffic to every computer in your network. By flooding a switch's ARP table with a ton of spoofed ARP replies, a hacker can overload many vendor's switches and then packet sniff your network while the switch is in "hub" mode.

Scared? Good, Now Calm Down!

This is scary stuff. ARP Cache Poisoning is trivial to exploit yet it can result in very significant network compromise. However, before you jump to Defcon-7, notice the major mitigating factor: only local attackers can exploit ARP's insecurities. A hacker would need either physical access to your network, or control of a machine on your local network, in order to deliver an ARP Cache Poisoning attack. ARP's insecurities can't be exploited remotely.

That said, hackers have been known to gain local access to networks. Good network administrators should be aware of ARP Cache Poisoning techniques.

Since ARP Cache Poisoning results from a lack of security in a protocol that is required for TCP/IP networking to function, you can't fix it. But you can help prevent ARP attacks using the following techniques.

For Small Networks

If you manage a small network, you might try using static IP addresses and static ARP tables. Using CLI commands, such as "ipconfig /all" in Windows or "ifconfig" in 'NIX, you can learn the IP address and MAC address of every device in your network. Then using the "arp -s" command, you can add static ARP entries for all your known devices. "Static" means unchanging; this prevents hackers from adding spoofed ARP entries for devices in your network. You can even create a login script that would add these static entries to your PCs as they boot.

However, static ARP entries are hard to maintain; impossible in large networks. That's because every device you add to your network has to be manually added to your ARP script or entered into each machine's ARP table. But if you manage fewer than two dozen devices, this technique might work for you.

For Large Networks

If you manage a large network, research your network switch's "Port Security" features. One "Port Security" feature lets you force your switch to allow only one MAC address for each physical port on the switch. This feature prevents hackers from changing the MAC address of their machine or from trying to map more than one MAC address to their machine. It can often help prevent ARP-based Man-in-the-Middle attacks.

For All Networks

Your best defense is understanding ARP Poisoning and monitoring for it. I'd highly recommend deploying an ARP monitoring tool, such as ARPwatch, to alert you when unusual ARP communication occurs. This kind of vigilance is still the greatest weapon against all kinds of attack -- for, as Robert Louis Stevenson wrote, *"The cruelest lies are often told in silence."*

Resources:

Address Resolution Protocol Spoofing and Man-in-the-Middle Attacks