2.3 ARP protocol and ARP spoofing

Spoofing

We mentioned a few types of attacks in the previous week. We'll look at them in more detail now starting with spoofing.

In a nutshell, spoofing is **pretending to be somebody else on the internet**. More specifically, since everyone on the internet is uniquely identified by an IP address and on a LAN network by a MAC address, **pretending to be someone else means tampering with the IP or MAC address**. Spoofing forms the basis of a lot of attacks because it is very easy to change these addresses.

As you recall, we are operating on a multi-layered network infrastructure. Each layer serves its purpose. In order to recognise the message sent inside the network, every layer has its own address to refer to.

Looking from the point of view of the network layer of the OSI model, we use an **IP address**. This represents a logical address. Logical addresses are used by networking software and socket interfaces to allow packets to be independent of the physical connection of the network, that is, to work with different network topologies and types of media. A logical address in the Internet used to be a 32-bit address (called IPv4 or IP version 4) that uniquely defined a host connected to the Internet. However, nowadays we are seeing a transition to the new standard IPv6 or IP version 6 that uses a 128-bit address. IPv6 was developed to deal with the long-anticipated problem of IPv4 address exhaustion.

Looking from the point of view of the link layer of the OSI model, we find what has been called a MAC address. MAC stands for **media access control** and represents a physical address which is the unique address assigned to a network interface controller (NIC). The address is hard coded on the NIC. The Ethernet header contains the MAC address of the source and the destination computer. This enables devices within a broadcast domain to send data back and forth. Most local area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, for example 07:01:02:01:2C:4B.

ARP protocol and ARP spoofing

So, in a multi-layered network infrastructure, we have different addresses. It has been necessary to create a set of protocols to link these addresses. For instance, for a given IP address, what is the physical destination address? And vice versa.

The two protocols created to solve these two problems are ARP and RARP.

The **Address Resolution Protocol** (ARP) is a communication protocol used for discovering the MAC Address associated with a given IP Address.

The **Reverse Address Resolution Protocol** (RARP) is an obsolete computer networking protocol used by a client computer to request its Internet Protocol (IPv4) address from a computer network, when all it has available is its link layer or hardware address, such as a MAC address. It has been rendered obsolete by the modern **Dynamic Host Configuration Protocol** (DHCP), which supports a much greater feature set than RARP. In all cases the client broadcasts the request and does not need prior knowledge of the network topology.

Let's see an example of how ARP operates. Let's say that system A is looking for the physical address of an unknown node whose IP address is: 141.23.56.23.

System A creates a request ARP message by providing the following information:

- the sender's physical address
- the sender's IP address
- the target's IP address
- the target physical address filled with zeros since it's not known.

The message is passed to the link layer where it is encapsulated in a frame which fills in the sender physical address and the missing target physical address with the broadcast address. The message is broadcasted. Every host or router on the LAN receives the frame. All stations pass it to ARP and all machines except the one targeted drop the packet.

The target machine replies with the ARP message that contains its physical address using a unicast message. The sender receives the reply message and knows the physical address of the target.

As we can see, the process is straightforward but a network can see a huge number of these requests. So to avoid having to send an ARP request packet each time, a host can cache the IP and the corresponding host address in its ARP table (ARP cache).

ARP is a stateless protocol. This means ARP continues to accept ARP replies and overwrite the old ones, even if they have not expired yet or even if the protocol did not ask them. Worse, ARP does not define any authentication method to check whether the replies come from the trusted one (the one we want to receive the replies).

The ARP cache and the fact that ARP is a stateless protocol offer chances for the attacker to perform ARP spoofing.

In short, ARP cache poisoning by spoofing works as follows. The attacker constructs spoofed ARP replies. A target computer could be convinced to send frames destined for computer A to instead computer B. The process is totally transparent to the target computer, so A will have no idea that this redirection took place. This process of updating a target computer's ARP cache is referred to variously as:

- ARP poisoning
- ARP spoofing
- ARP poison routing
- ARP cache poisoning.

Let's see how an attacker can perform a spoofing attack.

The main graphic is described below.

View description

Three computers connected to a switch. Computer A has IP 10.0.0.1 and MAC aa:aa:aa:aa, computer B has IP 10.0.0.2 and MAC bb:bb:bb:bb, and the hacker has IP 10.0.0.3 and MAC cc:cc:cc:cc. Below computer A there is the label 'ARP cache' with no values, below computer B there is the label 'ARP cache' with IP = 10.0.0.1 and MAC = aa:aa:aa:aa. The hacker sends a spoofed ARP reply with IP = 10.0.0.2 and MAC = cc:cc:cc:cc.

The attacker constructs spoofed ARP replies (the yellow box), where the attacker suggests that the IP address 10.0.0.2 that belongs to B, has MAC address cc:cc:cc:cc which is the attacker's MAC address.

The message is forwarded in the network.

The message reaches computer A.

Computer A updates its ARP cache associating the IP Address 10.0.0.2 with the MAC Address cc:cc:cc:cc. A's cache is poisoned and all packets that A intends for B get sent to the attacker.

Previous Next

An attack of ARP spoofing can face some difficulties. First of all, the ARP cache expires, so the entries need to be refreshed (about once every 40s is sufficient usually).

Man-in-the-Middle (MITM) attack with ARP spoofing

Let's look at how a Man-in-the-Middle attack is performed using ARP spoofing.

The goal of the attack is to insert the hacker's computer H in between the conversation of A and B, such that A and B should be able to continue their conversation, but with H having complete access to all their packets.

This is similar to the ARP spoofing attack we saw previously. The attacker will need to:

1. *poison T1's cache, spoofing T2's address as its own*
2. *poison T2's cache, spoofing T1's address as its own.*

This is the set up for the attack.

Then, when a packet comes through from T1 to T2, it goes to the hacker on the link layer, because of T1's poisoned cached. The attacker can examine this message and relay or copy it to T2, pretending to be T1. Similarly for messages in the reverse direction from T2 to T1.

The main graphic is described below.

View description

Three computers connected to a switch: computer T1 has IP = 10.0.0.1 and MAC = aa:aa:aa:aa; computer T2 has IP = 10.0.0.2 and MAC = bb:bb:bb:bb and the Hacker's computer has IP = 10.0.0.3 and MAC = cc:cc:cc:cc. The hacker sends a spoofed ARP reply with IP = 10.0.0.2 and MAC = cc:cc:cc:cc that reaches T1. Below computer T1 there is the label 'ARP cache' with IP=10.0.0.2 and MAC=cc:cc:cc:cc. The MAC address

is highlighted in red, to show it has been spoofed. Below computer T2 there is the label 'ARP cache' with IP = 10.0.0.1 and MAC = aa:aa:aa:aa.

The attacker spoofs the ARP reply and poisons the cache of T1.

The attacker also spoofs the ARP reply and poisons the cache of T2.

The attacker now receives messages meant for T2 before relaying them on.

The attacker also receives messages meant for T1 before relaying them on.

Previous Next

Switches have an internal table (CAM Table) which maps switch ports to MAC addresses:

2960-1#show mac address-table
Mac Address Table

Vlan	Mac Address	Type	Ports
1	00ld.70ab.5d60	DYNAMIC	2
1	00le.f724.al60	DYNAMIC	3

It is possible to attack the availability of switches by attacking their CAM table (Contents Addressable Memory table), which maps switch ports to MAC addresses (see field Ports and Mac Address). This is possible taking advantage of ARP messages. This can be seen as an ***active*** *technique for intercepting traffic.*

In a MAC flooding attack, a switch is fed many Ethernet frames, each containing different spoofed source MAC addresses to consume the limited memory in the switch (failopen mode) and force significant quantities of incoming frames to be flooded out on all ports (as with a hub), instead of just down the correct port as per normal operation.