ARP Spoofing Attack Lab

Contents

Lab Objectives and Setting

Overview

The Internet protocol suite, also known as TCP/IP suite, is the foundation of the modern Internet. This suite provides various protocols that lie upon the Internet Protocol (IP) for end-to-end data communications. Vulnerabilities in TCP/IP protocols may have serious effects on the upper layer protocols and applications and can endanger communication and data security. In this lab, we will look into ARP protocol, learn how it works and analyze its vulnerability, and show the way to attack it.

Objectives

Upon completion of this lab, students will be able to:

- Recognize the operation procedure and data format of ARP protocol;
- Explain how ARP spoofing attack works;
- Analyze the vulnerability of ARP protocol and give a feasible protection solution.

Prerequisites

- Practical experience with SSH and basic Linux commands;
- Basic knowledge of network protocols.

EZSetup

EZSetup is a Web application capable of creating various user-defined cybersecurity practice environments (e.g., labs and competition scenarios) in one or more computing clouds (e.g., OpenStack or Amazon AWS). EZSetup provides a Web user interface for practice designers to visually create a practice scenario and easily deploy it in a computing cloud, which allows for customization and reduces overhead in creating and using practice environments. End users are shielded from the complexity of creating and maintaining practice environments and therefore can concentrate on cybersecurity practices. More information about EZSetup can be found at https://promise.nexus-lab.org/platform/.

Environment Setting

In this lab, students can access three virtual machines (VM) from EZSetup, attacker, victim, and observer VM. The network topology is shown in Figure 1, the VM properties are listed in Table 1. Please refer to the EZSetup dashboard for the actual public IP addresses and passwords.

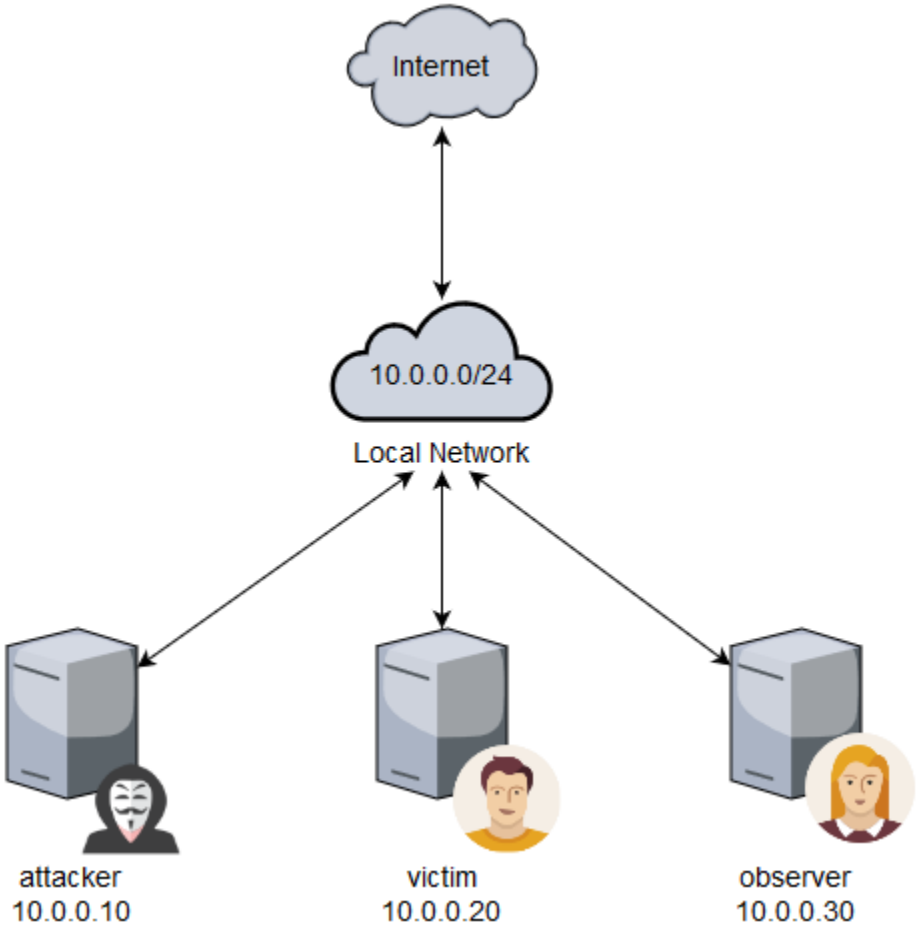

Figure 1 Lab network topology

Table 1 VM properties and access information

Name	Image	RAM	VCPU	Disk	Login account
ts-attacker	tcpipsecurity-attacker	2GB	2	40GB	See EZSetup
ts-victim	tcpipsecurity-victim	4GB	2	60GB	See EZSetup
ts-observer	tcpipsecurity-observer	2GB	2	40GB	See EZSetup

License

ARP Spoofing Attack

The Address Resolution Protocol (ARP) is an important link layer protocol for mapping upper layer protocol address to lower layer hardware address, such as IPv4 address to MAC address. ARP is usually used in host discovery within a single network, and it is not routed across different networks.

The message format of ARP is shown in Figure 2. The numbers above the table measure the length in bytes of a field, and the numbers on the left of the table give the offset of a field of the ARP packet header. The **hardware type** field specifies lower layer protocol type, e.g., hardware type for Ethernet is 1. The **protocol type** field specifies the EtherType value of upper layer protocol, e.g., 0x0800 for IPv4. **Hardware address length** field defines the length of following hardware address of sender and target, and **protocol address length** defines the length of sender and target's protocol address. **Operation** field has four possible values: 1 for ARP request, 2 for ARP reply, 3 for RARP (Reverse Address Resolution Protocol) request, and 4 for RARP reply. The content and size of following hardware and protocol addresses are decided by previous fields.

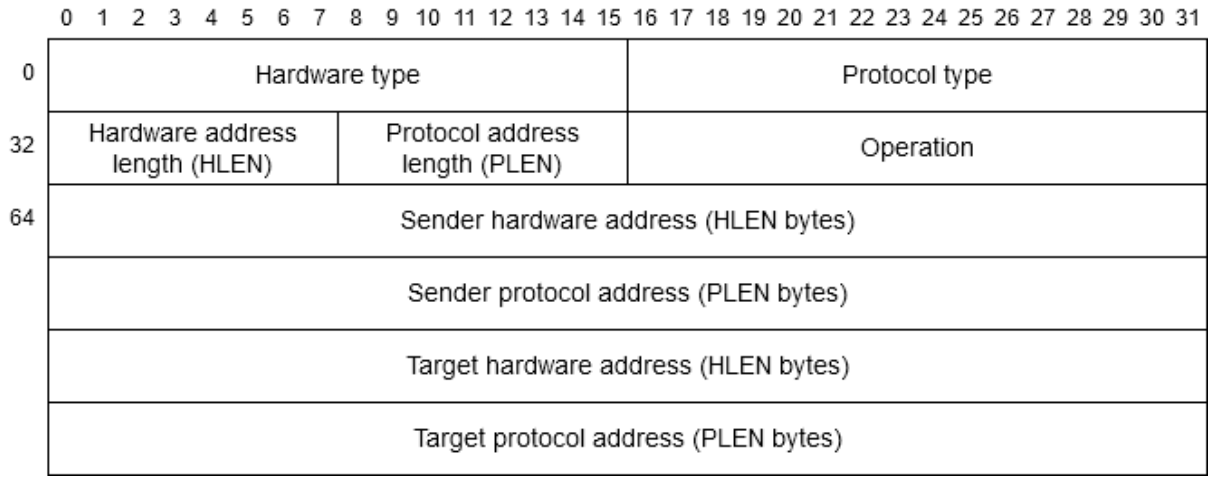

Figure 2 ARP message format

Let's assume the hardware type is Ethernet and upper layer protocol is IPv4. To find the hardware address of a target host within the same network for the first time, a host will broadcast an ARP request with its MAC and IP address, as well as target host's IP address to the network. At this moment, the target's MAC address is not known, so it is set to FF:FF:FF:FF:FF:FF. Then, upon receiving this request, the target host will reply an ARP response with its MAC address. Both hosts will also insert an entry in their ARP cache table with each other's IP and MAC address to save for future queries until the entry expires.

Because ARP does not authenticate replies on the network, forged ARP replies sent by a malicious host can be cached by the ARP request sender, even when it does not have the asked hardware and protocol address. As a result, the victim will send all of its traffic to a specific host to the malicious host, making the malicious host a "man-in-the-middle." Figure 3 shows the process of the ARP spoofing attack.

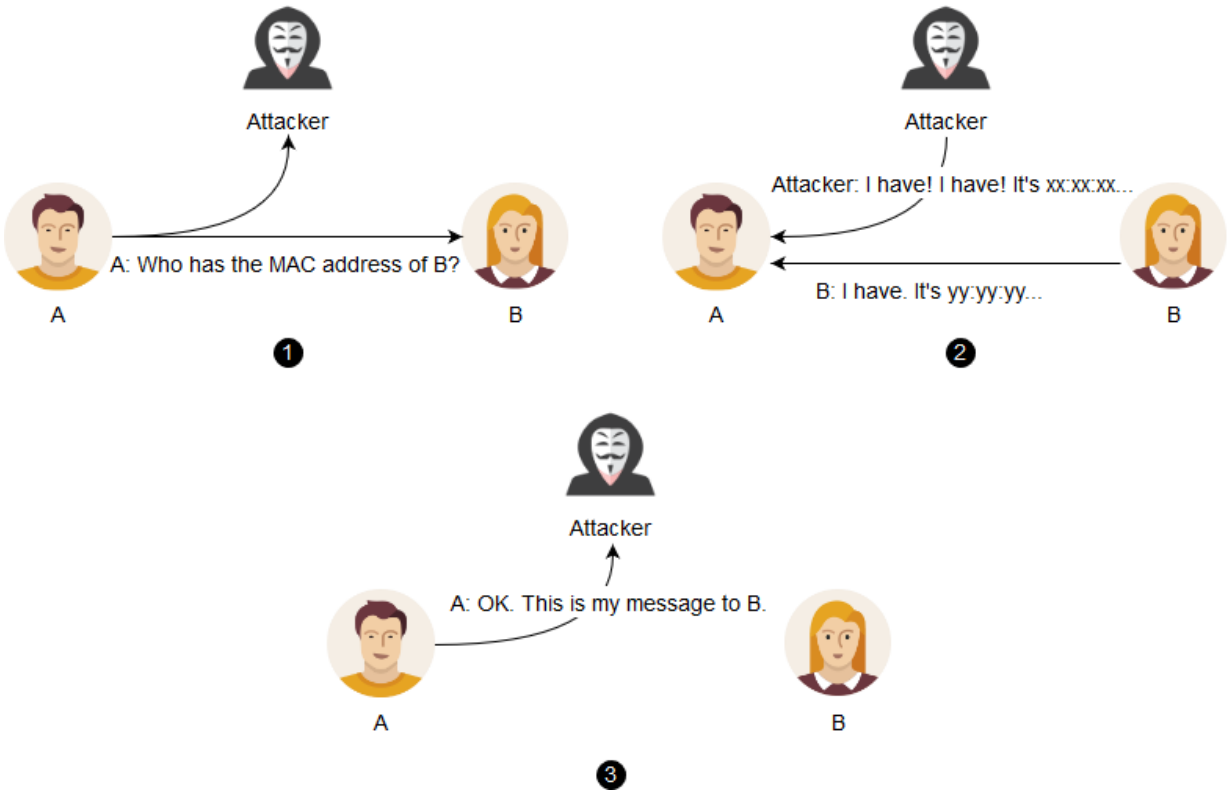

Figure 3 ARP spoofing attack process

Sometimes a machine needs to announce its ownership of an IP address on a network. It will send a "gratuitous ARP reply", which is an ARP reply but without a prior ARP request. The receiving hosts will update their ARP cache table to record this change. This is useful for a moving IP address because the IP address may be bound to different devices from time to time. And instead of other hosts asking for its MAC address, gratuitous ARP reply can be more efficient. But a malicious host can also exploit this to poison the ARP table of other hosts.

We can start an ARP spoofing attack using the arping tool. arping is a Linux command line tool for sending and receiving ARP requests and replies. To start an attack, use the following command to send gratuitous ARP replies to a victim:

$ sudo arping -q -c 3 -P -S <spoofed_ip> -I <interface> <target_ip>

The meanings of the options in the above command are listed below:

-q: suppress output
-c: number of packets to be sent
-P: send ARP replies instead of requests
-S: override the packet sender's IP
-I: network interface through which the packets will be sent

To view all the ARP entries in the cache table, please use this command:

$ sudo arp -a

To delete an ARP entry from the cache table, you can use the following command:

$ sudo arp -d <IP address>

Assignment

1. On attacker VM, send spoofed ARP replies to the victim VM with the IP address of the observer VM. Then, try to ping the observer from the victim, and capture incoming ICMP packets on the attacker. What command(s) do you use for this attack? How do you know your attack is successful?

2. Run the arpspoof tool from attack VM:

   $ sudo ./arpspoof <victim IP> <observer IP>

   This will carry out an ARP spoofing attack on both the victim and the observer. Then, start a simple HTTP server on the victim using:

   $ sudo python -m SimpleHTTPServer 80

   After that, try to visit the website hosted on the victim from the observer's browser. What do you see on the attacker's terminal? What does this mean?

3. Can you think of any way of preventing ARP spoofing attack?

Complete all the tasks and save your answer (with screenshots) to each of the tasks into a PDF file. Submit the PDF file.