Implementation of the ARP spoofing detection method from the paper "Detecting ARP Spoofing: An Active Technique"

Suggested by Vivek Ramachandran[1] and Sukumar Nandi[2], December 2005.

[1] Cisco Systems, Inc., Bangalore India
[2] Indian Institute of Technology, Guwahati, Assam, India

Implemented in **python** using **scapy** library

Yoel Bassin and Dov Greenwald, JCT, November 2020.

Note: this is not a full implementation of the algorithm suggested in the paper, since it will only alarm and will not stop the attack and drop the packets.

Summary

This detection method consists two main modules (we divided them into three parts, check_arp_header and known_traffic are the passive modules and spoof_detection is the active module).

1. The passive module:
   a. Check ARP header module:
      We check if the ARP message is inconsistent i.e. the Ether source / destination MAC address is not the same as the ARP source / destination MAC address. Detection of inconsistency guarantees the packets are spoofed, as such an anomaly is only possible in attack traffic, so we raise an alarm.
   b. Known traffic module:
      If the message is consistent, we will check if the IP and MAC pair in the request message are coherent with the <IP, MAC> pairs in our learned database, or raise an alarm if there are any contradictions. If the ARP packet is from an unknown IP address, it is sent to the active module.
2. The active module: Spoof detection module:
   We check if the ARP packet is an answer for an ARP request the machine sent in the threshold time (we chose 5 seconds).
   If the packet is an answer, we send a TCP SYN to the source <IP, MAC> of the packet. If we do not receive a TCP ACK, it indicates that we are under attack. However, if we receive a TCP ACK we will add the <IP, MAC> to the database.
   If the packet is not an answer, we will send an ARP request for the source IP of the packet, and we will recursively check the answers. Here there are two options:
   a. Both the attacker and the real host of the IP reply to the message. Thus, causing multiple replies for the same IP from multiple MAC addresses, so we may induce that an attack is occurring.
   b. Only the real host of the IP will reply to the message, and we will authenticate it via the TCP SYN message, so later when the attacker sends ARP replies the IP will be recognized and the message will be raising alarms (reference to 1. b).

Generally, we assume that the attacker will not reply to the TCP SYN and ARP messages, but if it does, we will get multiple replies and raise an alarm.