Exercise 12 – Network protocol attacks contd.

Notes –

1. Download and open the exercise pack, as usual.
2. **Please remember that you'll need to run python with root privileges for sniffing / packet sending to work.**
3. To send packets at layer 2, look in to the sendp() / srp() functions.
4. Aside from the programming assignments, please make sure to answer all the questions in the exercise and submit them (all at the root of your submission directory, like last exercise).

Questions –

1. ARP Ping –
    a. Using scapy, write a python script that performs an ARP "ping" - checking if a host is alive, by sending an ARP request for the host, and reporting the host as alive if a reply is received within the timeout.
    b. Input –
    Your script should comply with the following interface:
    arp_ping.py [HOST]
    e.g.: "./ex12_q1.py 192.168.1.1" will test whether the host at 192.168.1.1 is up.
    c. Output –
    If a reply is received – "[HOST] is up" (e.g.: "192.168.1.1 is up")
    If the timeout is reached- "[Host] is down" (e.g.: "192.168.1.1 is down")
    d. Answer this question in a file named ex12_q1.txt –
    Would this utility work over the internet? (e.g. – testing if www.google.com is available)
    Explain why.

    PLEASE SUBMIT:
    ex12_q1.py – your scapy script
    ex12_q1.txt – your documentation and answer to q1.d

2. Detecting promiscuous hosts –
    a. Assume you're on a network that is built using hubs only (every packet reaches every host on the network).
       Because this network is so insecure, you want to see if someone on the network is running a packet sniffer and monitoring packets in promiscuous mode.
    b. By creating an ICMP echo reply (ping request) packet whose IP destination address is the suspected host and whose MAC destination address is wrong (anything other than 00:00:00:00:00, FF:FF:FF:FF:FF and the valid address will do) – you can detect exactly this condition.
    c. Create a python script to test whether a host is in promiscuous mode as explained.
    d. Input –
       Your script should comply with the following interface:
       is_promiscuous.py [HOST]
       e.g.: "./ex12_q2.py 192.168.1.1" will test whether the host at 192.168.1.1 is in promiscuous mode.
    e. Output –
       If a reply is received – "[HOST] is promiscuous" (e.g.: "192.168.1.1 is promiscuous")
       If the timeout is reached- "[Host] is not promiscuous" (e.g.: "192.168.1.1 is not promiscuous")

       PLEASE SUBMIT:
       ex12_q2.py – your scapy script
       ex12_q2.txt – your documentation

3. ARP Poisoning –
    a. Write a python script to perform ARP poisoning on a target host in your network.
    b. Input –
       arp_poison.py [HOST TO ATTACK] [HOST TO IMPERSONATE]
       e.g.: "./ex12_q3.py 192.168.1.8 192.168.1.1" will make the host at 192.168.1.8 think that we (the machine who's running your tool) are in fact the host at 192.168.1.1.
    c. Output –
       None.

       PLEASE SUBMIT:
       ex12_q3.py – your scapy script
       ex12_q3.txt – your documentation

4. MAC Flooding –
    a. Run in another terminal: “sudo ./simple_switch_sim.py”
       This script will simulate a local switch. Your goal is to flood it with packets, so that its MAC table is filled and it is forced to fail-over to repeater (hub) mode.
    b. Write a python script to perform this flooding named ex12_q4.py.
    c. The switch simulator will help you know once you’ve successfully performed this attack.
       You may also use the mac_table_display.py script to see your attack in progress (run “sudo ./mac_table_display.py”)

       PLEASE SUBMIT:
       ex12_q4.py – your scapy script
       ex12_q4.txt – your documentation

5. Smurf Attack (ping reflection) –
    a. Write a python script to send a ping request to all the hosts on your network, whose source IP is spoofed to be the target host.
    b. Input –
       smurf.py [HOST]
       (e.g.: “./ex12_q5.py 192.168.1.8”, which will send a ping request to 192.168.1.255 from 192.168.1.8, which will cause every host to send a packet to 192.168.1.8, thereby DDoSing it)
    c. Output –
       None.

       PLEASE SUBMIT:
       ex12_q5.py – your scapy script
       ex12_q5.txt – your documentation