Network Security: Scanning, Sniffing and Spoofing

(lab adapted from https://seedsecuritylabs.org/Labs_16.04/PDF/Sniffing_Spoofing.pdf)

Objective:

1. **Understand how scanning, sniffing and spoofing work in practice**
2. **In the process, get familiarized with relevant tools i.e. Nmap, Wireshark and Scapy.**

Preamble:

One fine day, a gentleman in his early fifties walks into your room. You don't seem to have seen him before and ask what you can do for him. The gentleman asks if you know Agent X. You are now beginning to panic, maybe this guy is an enemy agent out to get you. As the man advances towards you fingering something in his pocket, you start to back off. And bang! this is what happens next https://i.dailymail.co.uk/i/gif/2017/11/giphy.gif

Oh man! You can't believe your eyes. The mask Tom Cruise sports in “Mission Impossible” is indeed real! Agent X had played a nice cool trick on you!

Getting to the point, Agent X tells that he is going undercover for a few weeks in Gabbar's organization. Some insider within the organization has given him some juicy information which he wants to act on. He managed to get a job as a data entry operator. He may not be able to login to high security machines, but he sure will be given access to a machine that is plugged into the LAN. Based on it, his main goal is to determine a machine which runs a special service on port 7777 for Gabbar (as told to him by the insider); identify it by its IP address; sneak into the server room (he plans to flick some one's access card for this) and locate it (too many machines, but thankfully all machines are labelled by their IP addresses); bring it down and remove/swap its hard-disk (which contains sensitive information). But the catch is, Gabbar needs this machine to be always on (since he may want to access it any time remotely) and hence the sysad put in place a mechanism, where an observer machine periodically pings the special machine every sec and in case it does not reply, the observer will raise an alarm. Bringing down the observer (physically) is not an option, since some or the other sysad is always near it 24/7 (Agent X is yet to learn DOS). How can Agent X accomplish above inspite of the challenge?

Guidance:

1. We will use 3 VMs for this exercise. To clone and configure such that they are on the same LAN, follow the instructions in Apendix A and B of the below document. You may want to name them as observer, attacker and victim (though you may not always use them as the name suggests, but in some exercises the name will help avoid confusion.) https://seedsecuritylabs.org/Labs_16.04/Documents/SEEDVM_VirtualBoxManual.pdf
2. All below questions assume you are using the seed lab VMs (downloaded from https://seedsecuritylabs.org/lab_env.html), results will not match if you are using some other VMs.
3. Wireshark is a popular sniffing tool, and easy to use. We will continue to use this, however, it is difficult to use Wireshark as a building block to construct other tools. We can use the python tool Scapy for this purpose, where lot of its functionality can be integrated within our own program to sniff, spoof as well as launch attacks.

Exercise 1: Scanning via nmap

References:
https://securitytrails.com/blog/top-15-nmap-commands-to-scan-remote-hosts
https://blogs.sans.org/pen-testing/files/2013/10/NmapCheatSheetv1.1.pdf

Guidance:

1. Go through above references and understand nmap usage.
2. You will use two Vms, attacker and Victim.
3. If nmap is not installed on the attacker VM, install it via "sudo apt-get install nmap" (needs root access).
4. On the attacker VM, use nmap to achieve the following. In the process, be sure to start a wireshark trace on the attacker VM to observe the traffic being generated by nmap to figure this out. Note some of this may take time, so wait.
    1. How many machines are up in the 10.0.2.0/24 network?
    2. How many TCP ports are open on Victim VM?
    3. How many UDP ports are open on Victim VM? (this can take long)
    4. What is the OS being used by the Victim VM?
    5. Netcat is a useful tool to open ports, you can use "nc -l 7777" to run a tcp service which listens on port 7777 on victim VM and then run nmap on Attacker and see if you can discover this service.

Exercise 2: Sniffing via wireshark and Scapy

Before, solving Agent X's problem, let us get familiarized with Scapy.

References:

1. https://scapy.readthedocs.io/en/latest/usage.html
2. https://blogs.sans.org/pen-testing/files/2016/04/ScapyCheatSheet_v0.2.pdf
3. https://danielmiessler.com/study/tcpdump/

Guidance:

1. Telnet from Observer VM to Victim VM. Run wireshark on attacker VM to sniff packets of this connection. Identify the password used by Observer to login to Victim VM from the trace. Note, the VMs are connected via a bus topology and hence you are able to sniff, normally in a star topology you cannot. This is to demonstrate the use of wireshark.
2. Go through and play around with the given scapy script sniff.py. This script sniffs ICMP packets. While the script is running on attacker VM, ping Victim VM from Observer VM and see if you can capture the packets.
   Note: Run the script with root privileges using sudo.
3. Extend the scapy script to capture any TCP packet that come from or go to "www.iitb.ac.in". Scapy's filter use the BPF (Berkeley Packet Filter) syntax; which is the same as what tcpdump also uses. See above reference for examples. To test the correctness, use the browser on Victim VM to contact some other website, your script (running on Attacker VM) should not print any packet. Use the browser on Victim VM now to access www.iitb.ac.in website, the script should now print these packets.

Exercise 3: Spoofing via Scapy

Guidance:

1. The sample scapy script spoof.py generates an ICMP packet. Run it on attacker VM . Set the destination address to one of the other VMs in the script. Check via wireshark if indeed the packet is being generated and sent and also whether you are recieving a reply in response to this.
   Note: Run the script with root privileges using sudo. Be sure to start wireshark before running the script.
2. Now extend the script to spoof Victim's IP address i.e. the attacker is sending an ICMP echo request to Observer VM pretending to be Victim VM. Run wireshark on Observer VM to see if each is receiving the request and reply respectively.

Exercise 4: Sniff and Spoof

Reference:

1. https://gist.github.com/ansipes/103cf69ab31f9e09fde0fe049c47bb54

Observer VM periodically pings Victim VM (which Agent X wants to bring down to swap the hard drive). To prevent Observer from raising an alarm due to unreachability, Agent X's machine needs to sniff the channel to monitor ICMP echo requests from Observer and then spoof the ICMP echo reply as if it is coming from Victim VM.

Guidance:

1. Build upon the earlier scripts to sniff ICMP echo requests from Observer VM to Victim VM and if you observer them, spoof a corresponding ICMP echo reply to trick the Observer. The above reference more or less reveals what is needed.
2. While you play with your script, you can keep the victim VM up (you may notice multiple replies). Once you are ready, do below steps.
3. Before start of the experiment, from observer ping victim, you should see that it is up according to ping (this step is important).
4. Then shut down Victim VM.
5. Start your script on attacker and ping Victim from Observer. You should ideally see no difference from what was observed in step 2. Note that if the Observer VM pings some other machine, say some external machine (e.g. www.iitb.ac.in), attacker should not reply, only requests directed to Victim VM should be replied to by attacker.
6. Just for fun, modify your script to try to spoof replies from machines outside the local area network. Try to ping an external machine (say www.facebook.com which doesn't replies, and www.cse.iitb.ac.in which normally replies).