UMBC Cyber Dawgs - Networking

By: Christian Beam

CTF PCAP AND QUESTIONS:

http.pcapng
1. Simple http tcp stream

This user is doing some research for a new pet they would like to get.

1. What search engine did the user use?
2. What did they search for?
3. How many packets were sent over the tcp port 80?
4. What website containing information about this pet did they visit?

Beginners:

1. Do pcap questions
    a. Download the http.pcapng from the github
2. Answer this: Say that I have network 192.168.0.0 and want to subnet it to support all my networks and hosts, and say that I expect to have 100 remote sites with 300 PCs each. What subnet mask should I use?
3. Open Wireshark, start capturing on your network interface card (NIC) that is connected to the internet
4. Open a browser and visit some pages (try to find one that is HTTP only)
5. Stop the capture
6. Go through the pcap and look around at the different packets for each protocol, what is different/the same?
7. Try applying some filters

Advanced:
1. Do PCAPS from last years' CTF

2. Answer this: Say that I have network 192.168.0.0 and want to subnet it to support all my networks and hosts, and say that I expect to have 100 remote sites with 300 PCs each. What subnet mask should I use?
3. Configure your vms
    a. Have one NAT interface, and one host-only interface on both hosts
    b. Install python pip on host0
        i. Sudo apt-get install python-pip
    c. Upgrade pip
        i. Pip install --upgrade pip
    d. Install scapy and cryptography packages
        i. Pip install scapy
        ii. Pip install cryptography
4. Give each an ip
    a. Host0
        i. sudo ifconfig <**HOST ONLY INTERFACE**> 192.168.1.10/24
    b. Host1
        i. sudo ifconfig <**HOST ONLY INTERFACE**> 192.168.1.11/24
5. Setup a listening port on host1 and run it in the background
    a. sudo nc -l <port> &
    b. Check netstat to ensure it is listening on that port
        i. sudo netstat -tulpn
6. Back on host0, we need to configure iptables not to drop a packet we are going to create and send to host1
    a. This is because we are only sending a single raw SYN packet which the kernel will try and block
    b. **iptables -A OUTPUT -p tcp --tcp-flags RST RST -j DROP**
    c. Make sure to remove this rule after this lab
7. We are now going to use SCAPY to create a packet and send it to host1
    a. sudo scapy
    b. You should have a >>> prompt now for scapy
    c. >>> i = IP()
    d. >>> i.dst = “<host1 IP>”
    e. >>> i.display()

        i. displays the current packet we have
    f. >>> t = TCP()
    g. >>> t.dport = <host1 listening port>
    h. >>> t.flags = “S”
    i. >>> t.display()
8. Now that we have our RAW SYN packet created, we will send it to host1
    a. >>> sr1(i/t)
        i. 1 is a numerical 1 not lowercase L
        ii. This should send the SYN to host1 who will send back a SYN-ACK and wait for the last ACK from host0
9. Practice crafting packets and sending them between hosts, you will spend a lot of time figuring out why or how these things will work

- You now have a great test bed for practicing Cybersecurity
    - You should have a snapshot of the clean host before you installed anything, you can keep taking snapshots as you add more standard packages that you will be using
- Use cases:
    - CCDC Service setup
        - Use these hosts to setup services, and test those services with a client
        - Figure out the necessary ports and protocols required by that service
        - Block all the rest that are not necessary (this is how we lock services down)
    - Attack and defend
        - **DO NOT HAVE A NAT INTERFACE IF YOU DO THIS, YOU DO NOT WANT TO TAKE THE CHANCE OF ACCIDENTALLY SENDING AN EXPLOIT TO THE INTERNET**
        - Use these hosts to exploit each other, or setup a metasploitable VM and exploit that
        - Learn how to defend from those attacks (i.e. what do you have to update, configure, or block to prevent it)

BONUS: USB PCAPS

1. Create a pdf and insert a usb stick
2. Start USB capture from Wireshark
3. Move it to the USB
4. Stop the capture
5. Use tshark to then pull out the pdf data from the packets in the capture and output the bytes to a file
6. Find a program that converts those bytes back into a readable pdf and feed it the file with the raw bytes