Lab 1

Due: September 11

Networking Tools

The objective of this lab is to learn basic network programming as well as familiarize yourself with useful command line tools. Be sure to answer all explicit questions from each section. You should turn in one pdf of your write up, any source code used, as well as any tcpdump files requested or referenced in the report. A LaTeX template can be found on Canvas. All files should be bundled in a zip or tar archive and submitted on Canvas.

Part 1:

Include in your write up: how your attack works, and how the recorded pcap of the attack relates to it.

Step 1: Create an echo server that capitalizes string inputs and a client program to interact with the server. **Both your client and server must be written in C**. The server's job is to capitalize the string sent by the client. Your program must behave exactly the same as the provided python client and server. This means you should be able to run the python server and interact with it using your client, as well as interact with your server using the python client.

Your server should be implemented in a single `server.c` file. Your client should be implemented in a single `client.c` file. Include both files as well as any necessary .h files in your final zip file. In your report, please describe how to build the server and client.

Step 2: Implement a Denial Of Service (DOS) attack on the server you wrote. A successful DOS attack should make it difficult or impossible for the client to connect to the server. To implement this attack you may have to disable SYN cookies on the host machine. Record a tcpdump of the attack in action.

Reminder: `sudo tcpdump -i any -w output.pcap`

Step 3: Extra credit. Use bro/zeek to monitor network traffic and detect the DOS attack. Include what you did and how your detector works in the write up.

Part 2:

1. Scan the Internet with zmap on port 80. Use the blacklist found at http://64.106.81.7/blacklist.txt. Limit your scan using the -t flag in zmap so that your scan only lasts 2-4 hours. "-t NUMSECONDS".
2. Once the scan is complete, systematically find the subnets of each network and group together IP addresses in the same network. Use the `whois` tool.
3. Create a report of the results. Include useful information such as, but not limited to: Number of IPs per network, number of machines responding / number of machines probed owners.
4. For your report, dig in deep into one network. Report what you learned about the network using command line tools. Do not use `nmap -A`, it may be seen as an attack. You may slowly scan with zmap using other common ports such as 22 or 443.

Part 3:

Access the 10 websites below, 10 times each, for 3 different connection modes: VPN, TOR browser and regular browser (firefox, chrome, safari...). For each access of a site, be sure to record the network traffic with tcpdump.

For each tuple (connection type, website), plot statistics about the connection, e.g. average packet size, average number of packets sent etc.

Report your findings, (you do not need to include all the graphs in the report). Answer the following questions and include any other important discoveries.

- For each connection type, what is visible to a passive device on the network?
- Can you use the connection statistics to determine which of the 10 websites was visited?

Note 1: You may find tshark helpful to print packet information to the command line, or to an ascii file (`tshark -r test.pcap > readable.txt`).

Note 2: If you don't have access to a VPN, you may use the campus VPN (https://wikis.utexas.edu/display/eceit/UT+VPN)

Websites:

1. https://en.wikipedia.org/wiki/Cat
2. https://en.wikipedia.org/wiki/Dog
3. https://en.wikipedia.org/wiki/Egress_filtering
4. http://web.mit.edu/
5. http://www.unm.edu/

6. https://www.cmu.edu/
7. https://www.berkeley.edu/
8. https://www.utexas.edu/
9. https://www.asu.edu/
10. https://www.utdallas.edu/

Part 4:

Record all steps of this section using tcpdump. For this section, if you are using a virtual machine you must take the tcpdump from the host machine or the TTL viewed in the packet capture will not be correct.

1. Connect to the website http://www.baiwanzhan.com.
2. Enter in the search bar any word.
3. Enter in the search bar the word

   法轮.
4. Use python to split the GET request up so that the word

   法轮

   is split up between two packets. Make sure your code that splits works with benign data as well as the word that causes a RST to be sent.

Examine the packets from steps 2 and 3.

- Open the pcap file with wireshark.
- Find the packet containing the GET request using the magnifying glass icon to search for the benign word from step 2 above.
- Secondary mouse click on the GET request packet and select follow TCP stream.
- Observer the Time to Live (TTL) field in the packets sent from the server from steps 2 and 3 (you may have to clear the search in wireshark to find the packets associated with step 3). Write in your report why you think the TTLs from steps 2 and 3 have the values they do.