EEw382N Security Laboratory Exercise 1 Report

Student: Your Name and EID Here
Professor: Mohit Tiwari
TA: Austin Harris
Department of Electrical & Computer Engineering
The University of Texas at Austin

September 14, 2019

1 Problem 1

Step 1: Please see server.c and client.c for the echo server construction and message processing source code.

Step 2: The DOS attack was implemented by putting the connect command in a while loop in order to keep sending the SYN signal to server. As a result, the server is flooded with SYN (Figure 1), therefore the client is unable to establish three-way handshake to the server.

The communication for localhost connections (anything that uses 127.0.0.0/8) go through what is called the loopback interface. Meanwhile to record the pcap package log, the following command is used with flag -i set to lo:

sudo tcpdump -i lo -w output.pcap

2 Problem 2

Part 1: For the first part, I ran the following command to scan the blacklist ip addresses: "sudo zmap -p 80 -t 7200 -b blacklist.txt". Please see results.csv to see the probed ip addresses.

Part 2: The python script used for grouping and finding CIDR subnet is named subnet.py. The grouped result is stored in grouping.txt

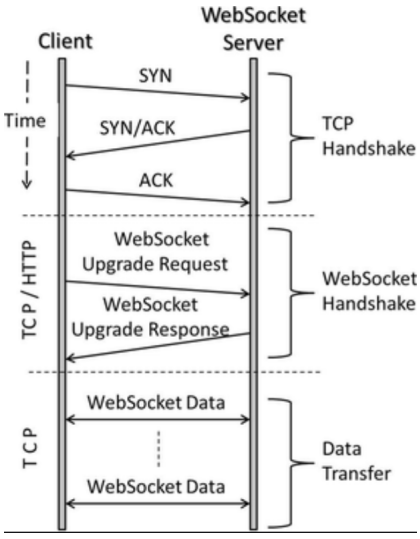

Figure 1: The attack.c keeps sending the SYN shown on top of the figure

Part 3: During the 2 hour zmap scan, 4,312,548 machines were probed, however, only 822 machines responded, and the hit rate is 0.04. All the responded ip addresses are stored in the results.csv. After running the grouping subnet.py function, which ran IPWhois() function to find the CIDR addresses, then grouped CIDR with the same IP addresses. Please see grouping.txt for more detail.

Part 4: For this part, I digged into the following ip address: 150.60.87.26, which has the following subnets: [150.26.0.0/15, 150.28.0.0/14, 150.32.0.0/11, 150.64.0.0/12, 150.80.0.0/13, 150.88.0.0/14, 150.92.0.0/15, 150.94.0.0/16, 150.95.0.0/20]. It appears that it is the IP address for Osaka University. I digged deep by running whois on the ip address, and found out the inet ranges from 150.60.0.0 - 150.60.255.255, which matches the CIDR subnet result returned from IPWhois. Also, we know that the IP is administered by APNIC (Japan Network Information Center)

3 Problem 3

Make sure the selenium package is installed: pip3 install selenium

Download the latest geckodriver for Firefox:
https://github.com/mozilla/geckodriver/releases

Make sure to confirure the PATH to gechodriver: export
PATH=$PATH:/home/class/Desktop/HW1

Following is the diagram for average packet size, average number of packets sent for VPN, TOR, and Firefox browser:

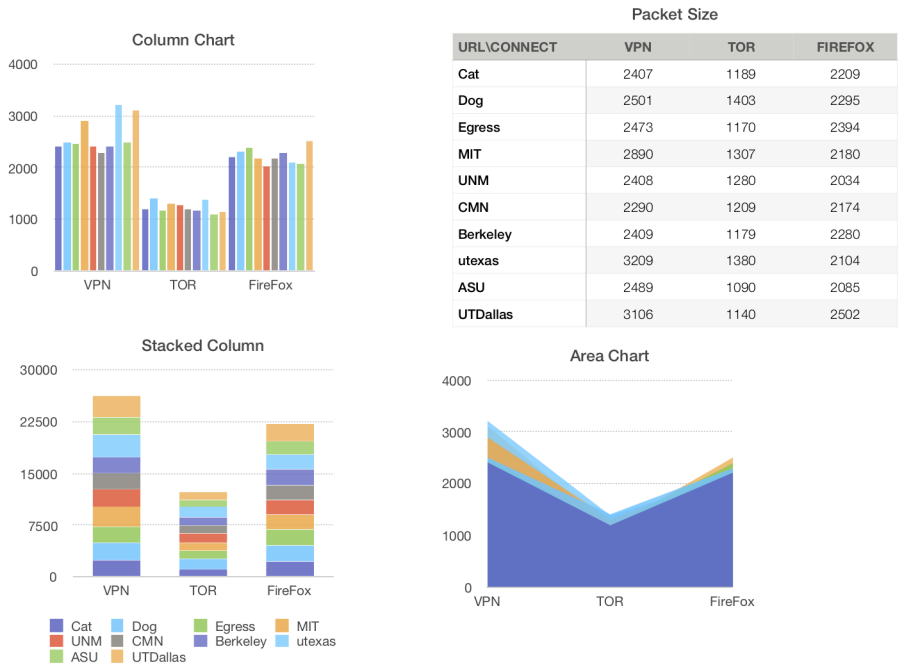

Figure 2: Average packet size sent with different connection types

We found that the average packet size for TOR connections are smaller than that of VPN and generic FireFox browser. Also, the number of packets sent using TOR is also less than VPN and FireFox.

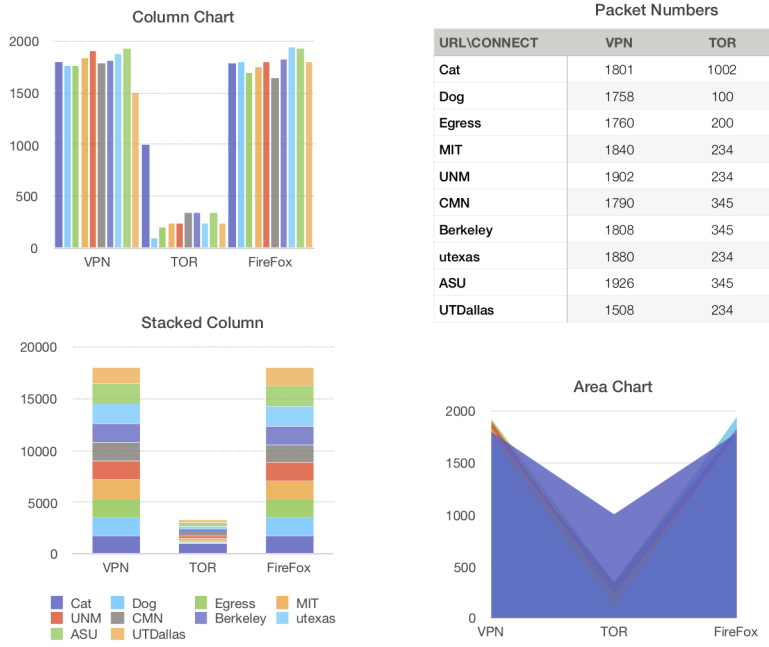

Figure 3: Average packet number sent with different connection types

For each connection type, what is visible to a passive device on the network? For generic connection using browser, the passive machine can see the encrypted message, sender and reciever addresses. For VPN connections, the passive machine can also see the encrypted message. However the sender address is replaced by the intermediate VPN ip addresses. For TOR connections, we cannot see ant encrypted message from the packets being sent, as a matter of fact, we could only see the packets for connecting to the TOR nodes.

Can you use the connection statistics to determine which of the 10 websites was visited? For some of the websites such as the Utexas and UTDallas websites, the average packet size is larger that those from wikipedia websites. Therefore, we can roughly tell the from the statics about the visited website.

4 Problem 4

From the follow TCP stream $tcp_1.txt$ $tcp_2.txt$ files we can see that the TTL for bengin word search is longer than search for the other word. The reason why TCP packets from step two has longer TTL than that of step three is that the baiwanzhan website completed the hand-shake process and returned the requested information. However, when we search for the second keyword, the website refused connection and did not return any packets due to unkonwn reasons. After some research I found that the word "" is actually the title of a government banned cult in China, therefore any search for those two words are banned.

To split up the GET request, we used Python socket library to send two queries to the baiwanzhan.com. After inspecting the wireshark packets, two TCP [PSH, ACK] packets has the first half of the query and the second half of the query. Which means the python script successfully splitted the query into two portions.