Malik Elgomati

Cyber/Infrastructure and Defense

Due April 14, 2023

DDoD Simulation Report

Throughout this assignment, I had the opportunity to learn new topics and skill sets such as using NS and learning to code .tcl files through Visual Studio Code. At the outset, was given a sample.tcl file and tasked with adding onto it to fulfill the requirements of the assignment's topology. To achieve this, I incorporated seven routers into the code, along with 12 bots that corresponded to these routers as well as mapped a total of four users into the network topology. Two bots were assigned to each router, with the exception of routers one and two, which only had one bot assigned each. The maximum traffic bandwidth for each link in the network was 950kb, as previously stipulated in the sample file. To effectively execute the attack, I decided to use three different target links, with each user having a designated target link. This allowed me to avoid any type of bot detection, as each link had its own respective bots attacking the web server. I opted for a calculated approach, which involved trial and error to identify the most effective way to attack the server. If I only implemented a single target link with seven routers and twelve bots, the attack would have likely been detected, thus decreasing the total degradation ratio.

For the first target link, I assigned routers 3, 5, and 6 to the webserver and bots 2, 5, 10, and bots 9 respectively. 5r bots were set to a rate of 250 kb, while the user was set to 100 kb. My aim was to overload the network bandwidth early on such that the user's 100 kb rate, which entered after one second of the attack,in order to cause the majority of its packets to be dropped.

For the second and third target links, I used the same approach, with four bots sending traffic at a rate of 250 kb and users sending traffic at a rate of 100kb. The amount of packets dropped for the rest of the users were similar, resulting in similar degradation rates.

For the last target link, I reverted to the original approach of having two bots and one user send traffic over the server, with the bots sending out 500 kb, and the user still sending out 100kb. This approach produced the lowest degradation percentage and wasn't as successful as the approach used for the other target links. However, I opted not to add more bots to the topology and kept the same approach.

To analyze the trace file, I used Wireshark, which I had prior experience with in Introduction to Networks. I opened the out.tr file in Wireshark, opened the statistics menu, selected the 'capture file properties' and then followed by the summary tab in order to pull out the specified user data I was interested in. I identified the 'packets dropped by capture filter' and 'packets dropped by interface' fields to indicate the packets dropped from each user every second. My results for the total amount of packets dropped are as follows:

User 1 dropped 198 packets out of 585 packets, which is a degradation percentage of 33.8%.

User 2 dropped 174 packets out of 585 packets, which is a degradation percentage of 29.7%.

User 3 dropped 195 packets out of 585 packets, which is a degradation percentage of 33.3%.

User 4 dropped 205 packets out of 585 packets, which is a degradation percentage of 35.0%.

Out of the total 2,340 packets, 772 packets were dropped resulting in a total of 32.9%

Figure 1 below shows my network topology for this simulation.

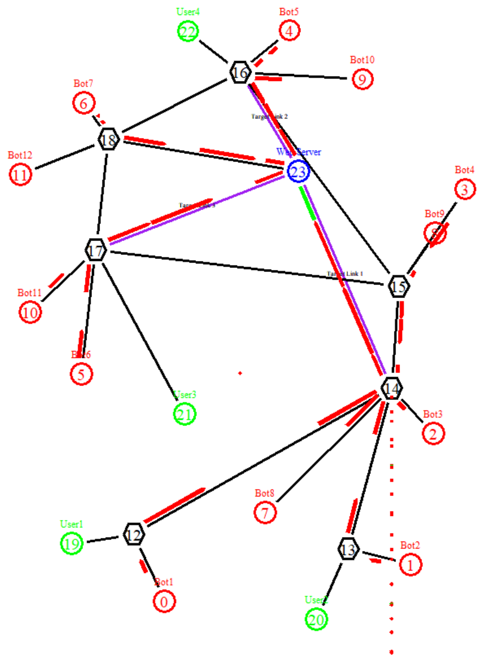

Figure 1. DDoS Simulation Network Topology

Figure 2 below shows the packets dropped against time from each user as well as the total packets dropped.

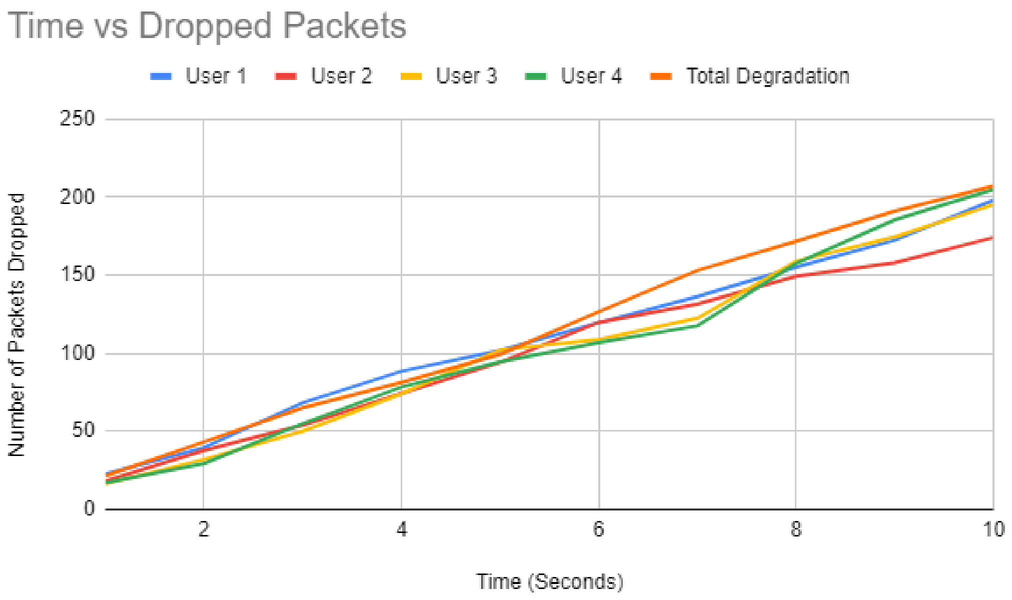

Figure 2. Time (Seconds) vs Number of Dropped Packets