Author: Temur Kholmatov
Group: B17-05
Course: Networks

Report

Running Node

Compile node_v2.c file: *gcc -pthread node_v2.c -o node*
Run: *./node*
The program will ask you to enter the ip and port of the first peer. Write it in the format “ip:port”. In case you are the first node of the net, write a word “first” and the program will just sit waiting for a sync from some other node.

Also, the program will prompt a message saying that you can input a filename to search for a particular file. In case the file is found in your own machine (your current directory), the program will read the file from your local storage. Otherwise, it will search the filename through its knowledge base of other peers. Finally, if the file is not found anywhere, it will prompt a related message.

Running DOS Attacker

You need to have python 3.* installed.
Run: python dos_attacker.py
The program will ask you to input the ip and port of your victim in a format “ip:port”. Afterward, it immediately initiate the DOS attack.

Attacking the Node

The DOS attack is performed in the way our TA suggested:

1. Connect to the peer.
2. Send sync flag - 1.
3. Send a greeting card. Notice I do not try to send fake greeting cards even if I could in order to make the attack more unpredictable. This kind of approach would count almost a cheating and disrespect to others.
4. Do not close the connection and go to step 1.
5. Also, I put a socket option SO_KEEPALIVE to 10 sec. However, I’m not sure it helps somehow.

Defending the Node

Even if I do not send fake greeting cards, some other person could do it to me. That is why I decided not to follow the suggestion of our TA about what data about a peer to hash for hashtable storage. I came to an idea of getting an ip address of the peer from the socket’s file descriptor and use it as the key.
Another feature I added is checking how many bytes are received from the peer. Usually, when the attacker interrupts sending data, the socket_rcv function in my node gets zero as a number of peers the connecting node knows. That is why I check if the number of bytes received is equal to zero. Then, I say that this peer did not send the full data and I won’t decrease its cdb counter. So, the cdb counter will remain increasing and, finally, after 5 connections this node will be put in the blacklist.