IS F462: Network Programming

Q3: Syn Flood

Design

The symflood program uses three processes, one to perform the actual attack, while the other two to listen on the loopback adapter, and ethernet adapter respectively using libpcap for incoming SYN and outgoing SYN+ACK TCP packets respectively. To perform the synflood, first a raw socket is opened. Then, IP and TCP headers are constructed, such that the TCP header has no payload. Finally the checksums for both are calculated and added to the headers. Once the header creation is completed, it is included as a header to any packets sent using the raw socket. This is the flow associated with the creation of one valid sym packet. To perform a flood, this is carried out in a loop, which is triggered using an alarm of 1 second. Thus, every second the server receives a syn packet breaing a random (valid) IP in its IP header, and a random (valid) port number in its TCP header, which the server attempts to acknowledge. However, since there is no client with teh given IP:PORT which actually generated the request, teh server never receives any ACK for its SYN-ACK, thus resources for the previous connection remain consumed.

Assumptions

The server is assumed to be running on the local machine, but the provided IP should be the ethernet IP address (the program worked sometimes on loopback IP, and failed at other times). The program will successfully perform the synflood for any server, but libpcap cannot listen for ACK's sent by the server if it is running elsewhere.

Execution Details

1. Compile the code using 'make'. This creates an executable called flood.out
2. Compile the provided minimal TCP server by running 'make server'. This creates an executable called server.out
3. Run the server first as follows:
   ./server.out <port_number>
4. Run the flood program with root priviledges as follows:
   sudo ./flood.out <hostname/IP_address> <port>

NOTE: If the server is not launched first, the program will display that it received RST instead of SYN+ACK.