Bar Naim 308045632
Aviv Yonai 204728125

Tftp_server.py Readme

The Program is creating a different UDP sockets for every request it receives, when the socket that receives those request is ORIGIN_SOCKET which binds to the input port, and receives initial packets. It does not terminate by itself. All the others sockets are terminated when the process is finished.

The Program maintains at all times:
SOCKET_DICT - a dictionary of socket: SocketInfo
READ_LIST - the first list we pass to select
WRITE_LIST - the second list we pass to select

Class SocketInfo:
Maintains the required data for each socket.

Program Flow

- The Program main Loop is initiated after ORIGINAL_SOCKET is bind and on READ_LIST
- The Function **establish_connection(data)** is called when select returned ORIGIN_SOCKET:
    - creates a random TID (between 3000-65535)
    - unpacks the first RRQ/WRQ Packet
    - Returns the kind of process, the new random TID and the file_name
- A new socket is created which binds to the address (clients_ip, TID)
- The SocketInfo of the new socket is updated
- The new socket is added to SEND_LIST
- The send process calls **handle_send(socket)** function which handles the send of each ready-to-send socket according to its SocketInfo, and passes the socket to READ_LIST
- When a socket is sent a timer (which is part of its SocketInfo) is initiated.
- The receive process calls **handle_receive(socket)** function which handles the receive of each ready-to-receive socket according to its SocketInfo, and passes the socket to WRITE_LIST
- When a socket is received the timer in its SocketInfo is canceled.
- After every socket is being handled by the above functions it is checked if that socket is finished and if so it closes.

Our Emails: avivyonai24@gmail.com, bnaim03@gmail.com