Lab1 - Information Flow Control

Aleksandar Mitic, Mohammad-Ali Omer

April 2020

1 Solution

1.1 Design of server

The server is always listening for a ("NEWPROFILE", data) message which holds all the information about a new dating-client. When a new profile is received the server tries to match it against all previously received profiles. Every profile contains an agent function which evaluates if a profile is suitable for itself. With the new profile we run this agent function against all other profiles and let the other profiles do the same with the new profile. If both profiles like each other then the server will send both clients each others profiles. Then the new profile is added to list of profiles on the server and the server starts listening for new profiles again.

1.2 Approach

Our approach to handling the information flow control properly was to step through our server code and tracking the PC and BL to look for false positives where a declassification could be applied safely. We landed on needing declassification for only the preference booleans and outer labels as specified in the lab description. We also lowered lowered the BL when sending out the profiles with pini.

Our approach for the malicious client was to send a initial profile which accepts all profiles in order to increase the chances to get a match. For each profile it matched with it would hijack that profile and resend it as it's own to match with some other profiles we could not match with initially. This process is repeated for each new profile it matches with. To avoid infinite loops we keep track of all the profiles we managed to leak so far to not send it back to the server again and avoid infinite matches.

2 Contributions

We feel like both contributed to the lab equally. For the most part we did pair programming. We sat together to set up the initial serve code accepting the in-

coming profiles and to get familiar with Troupe. We then split up, Mohammad wrote the base for the client-code and Aleksandar wrote the base for the matching algorithm. We continuously communicated with each other to make sure that both of us knew what the other one was doing and we debugged both the bening client and the matching algorithm together. We completed the rest of the lab using pair programming including setting up the p2p-network, handling the information flow correctly and writing the malicious client.