UNIVERSITÀ DI PISA

Computer Engineering

Foundations of Cybersecurity

secureCom

Group Project Report

TEAM MEMBERS:
Francesco Iemma
Yuri Mazzuoli
Olgerti Xhanej

Academic Year: 2020/2021

Contents

Chapter 1

How To Handle The Chat Request

Let's start from the simple case in which a client wants to chat with another client and makes a request to the server:

- The standard channel is used to send either the request for chat and for the server's answers.
- If the target client accepts then the main process receive the confirmation from the server, then the main process set isChatting to true and it starts to chat.

When the server receive the command to chat with someone he has to send the chat request to the given client. Let suppose that Alice wants to chat with Bob. In this case the following steps must be performed:

- The request must be done through the request channel
- The client daemon process (Alice's daemon process) is listening on its socket (see after) and receive the request, then it reads the variable isChatting: if it is true the daemon process refuses automatically the server request, otherwise it ask (HANDSHAKE) to the main process if he wants to speak with Bob.
- The main process answer to the server by means of the daemon tools, if the answer is positive then it sets isChatting to true and so it waits for the message from Bob.

It's important to underline some concepts, first of all on the server side we have one process that handles two socket with the client, one for each channel. The protocol starts with the client that contacts the server, then the server and the client main process establish a connection and starts the protocol to establish a secure communication (Key Exchange). For standard messages this standard channel is used.

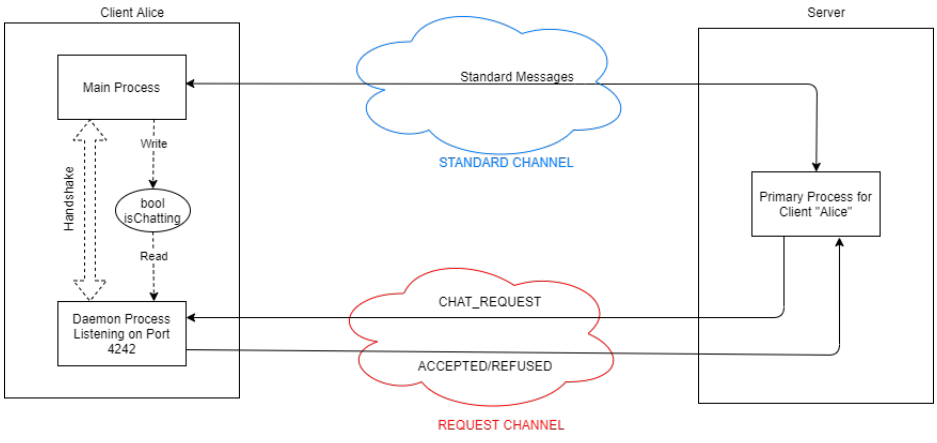

Figure 1.1: Protocol Schema

When the server has to sent a chat request to this client the request channel must be used. Thus it establishes a connection with the daemon process of the client that works as a server process and it is listening on port 4242 (see figure 1.1). The security of this secondary channel is ensured by the fact that the messages sent by the server to the client on the request channel are encrypted with the shared key established during the handshake in the standard channel, hence we can say that the authentication problem is not present. In any case to implement a greater security is possible to generate client side a one time password that is sent to the server in the encrypted session through the standard channel, then the server will sent this otp to the client daemon process to identify itself.
Other things to take in mind are:

- We assume that who send the chat request is the first to send messages.
- On the server side for each client two socket must be established, one for the standard channel and another one for the request channel but the process for each client is only one.