Trabalho de Redes de Computadores

Ana Laura Tostes
Eng. de Controle e Automação
Escola Politécnica - UFRJ
Rio de Janeiro, Brasil
Analauratostes.20221@poli.ufrj.br

Catarina Assunção de Carvalho
Eng. de Controle e Automação
Escola Politécnica - UFRJ
Rio de Janeiro, Brasil
rinasniw.20221@poli.ufrj.br

Gustavo de Oliveira Freitas
Eng. de Controle e Automação
Escola Politécnica - UFRJ
Rio de Janeiro, Brasil
frtgusta.20231@poli.ufrj.br

Samyla Nascimento Silva
Eng. de Controle e Automação
Escola Politécnica - UFRJ
Rio de Janeiro, Brasil
samyla.nascimento.20221@poli.ufrj.br

Thiago Mendonça Carneiro Morgado
Eng. de Controle e Automação
Escola Politécnica - UFRJ
Rio de Janeiro, Brasil
thiagomcm@poli.ufrj.br

***Abstract*—Este trabalho tem como objetivo descrever o protocolo escolhido pela turma para uma aplicação pensada e desenvolvida na disciplina Redes de Computadores. A aplicação consiste em um chat aberto para troca de mensagens entre alunos da disciplina.**

***Index Terms*—Chat, Cliente, Servidor, Socket, TCP.**

I. Introdução

A aplicação acordada compreende um chat desenvolvido em Python que permite a troca de mensagens entre sistemas finais conectados em uma mesma rede. Com base nisso, a turma produziu e aderiu a um protocolo condizente com a aplicação selecionada.

II. Aplicação

A interface escolhida pelo grupo é uma página de login e senha e uma parte destinada para enviar a mensagem.

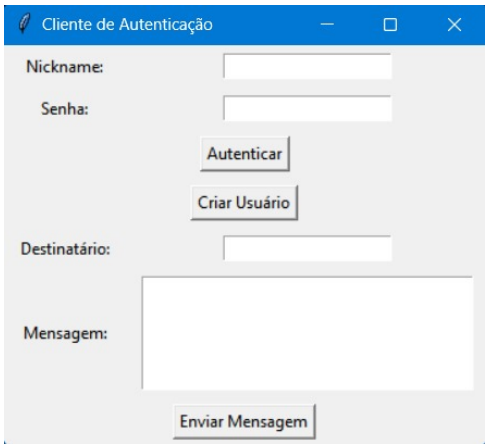

Fig. 1. Interface de autenticação desenvolvida em execução.

A interface é responsável por coletar a mensagem digitada no campo de texto e enviá-la. Além disso, a aplicação opera em conjunto com um terminal que registra as mensagens enviadas por todos os usuários conectados a partir da inicialização da aplicação.

Agradecemos ao apoio da instituição e ao nosso orientador, Prof. X.

A. Servidor

O servidor gerencia conexões com clientes utilizando sockets TCP na porta 7444. Quando uma conexão é estabelecida, ele aguarda o recebimento de uma mensagem criptografada, que é então descriptografada para processar a solicitação do cliente. A comunicação entre o servidor e os clientes é segura, utilizando criptografia baseada em uma chave gerada por um hash SHA-256 de uma string fixa.

Dependendo do valor da chave ‘flag‘ na mensagem, o servidor executa diferentes ações. Se a ‘flag‘ for 0, o servidor valida o nome de usuário e a senha do cliente. Se for 1, ele lida com o envio de mensagens entre usuários, armazenando as mensagens no arquivo JSON do destinatário. Caso a ‘flag‘ seja 3, o servidor cria um novo usuário, verificando se o nickname é válido.

Após processar a solicitação, o servidor criptografa a resposta antes de enviá-la ao cliente. As mensagens de usuários e as informações de autenticação são armazenadas em arquivos JSON, garantindo que as interações sejam salvas de maneira persistente. A criptografia de ponta a ponta garante a segurança da comunicação em todo o processo.

B. Cliente

O cliente interage com o servidor por meio de uma conexão TCP, utilizando o IP e a porta configurados. O cliente envia requisições criptografadas com dados de autenticação, criação de usuário ou envio de mensagens, dependendo da ação solicitada. Para isso, ele utiliza a biblioteca ‘cryptography‘ para criptografar os dados antes de enviá-los ao servidor. O cliente também recebe respostas criptografadas do servidor, que são então descriptografadas e apresentadas ao usuário.

A interface gráfica do cliente é desenvolvida com ‘tkinter‘, permitindo ao usuário inserir seu nome de usuário e senha, criar um novo usuário, ou enviar mensagens para outros usuários. Quando o usuário tenta se autenticar ou criar um

usuário, o cliente envia uma requisição com as credenciais fornecidas e espera a resposta do servidor, exibindo-a em uma janela de diálogo. Caso o usuário deseje enviar uma mensagem, ele preenche o destinatário e o conteúdo, e o cliente envia esses dados ao servidor, novamente aguardando a resposta.

Todo o processo de comunicação é realizado por meio da função sendrequest, que envia a requisição criptografada ao servidor e lida com a resposta. A comunicação é segura, pois os dados são criptografados antes de serem transmitidos e descriptografados ao chegar ao destino. O cliente tem controle total sobre as operações, interagindo de forma simples e direta com o servidor por meio da interface gráfica.

III. Transporte

A comunicação entre o cliente e o servidor ocorre por meio do protocolo TCP, que garante uma transmissão confiável e ordenada de dados. O protocolo de transporte assegura que as mensagens enviadas entre o cliente e o servidor cheguem corretamente, sem perda de pacotes, o que é fundamental para o funcionamento do sistema de chat.

IV. Rede

O sistema de chat utiliza a rede TCP/IP para estabelecer a comunicação entre os clientes e o servidor. A comunicação ocorre dentro de uma rede local ou pela Internet, dependendo da configuração do servidor. A escolha do protocolo TCP é justificada pela necessidade de garantir a integridade e a ordem na entrega das mensagens.

V. Conclusão

Este trabalho apresentou a implementação de um sistema de chat simples, porém seguro, utilizando criptografia para garantir a privacidade das mensagens trocadas entre os usuários. O uso de TCP como protocolo de transporte e a criptografia simétrica asseguram que a comunicação seja eficiente e confiável, com a integração da interface gráfica proporcionando uma experiência de uso amigável.