

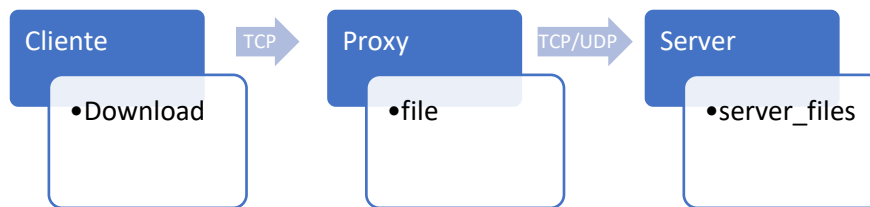
Projeto IRC

2019/2020

Bruno Faria 2018295474

José Reis 2018285575

Neste projeto pretende-se implementar uma solução de transferência de ficheiros, de uma pasta no servidor para a pasta de “download” num cliente. Esta transferência pode recorrer tanto ao protocolo TCP como ao protocolo UDP. É ainda implementada uma proxy que tem o objetivo de mediar as comunicações, além de poder introduzir perdas nos dados e intercetar estes dados guardando-os. Deve ainda ser possível encriptar os dados recorrendo à biblioteca Libsodium.



Começando pelo proxy, este media as ligações. Ao iniciar o proxy, este vai correr num endereço IP que lhe foi atribuído e na porta especificada como argumento, começa por criar uma ligação TCP da qual espera clientes. Após a ligação de um cliente, este cliente vai especificar o tipo de ligação entre o proxy e o servidor (TCP/UDP). Esta ligação proxy-servidor começa por ser TCP e após transmitir ao servidor o tipo de ligação pretendida, estabelece-se então a conexão determinada pelo cliente.

Sendo assim, o cliente ao ser iniciado tem como argumentos os IP's dos servidor e proxy além de um porto, que servirá para fazer a ligação com a proxy. Quando o cliente é iniciado envia uma mensagem à proxy com o endereço do servidor e o tipo de conexão pretendida. Após esta ligação o cliente está pronto a enviar comandos.

Analisando agora o servidor, este ao ser criado vai ter como argumentos o seu porto e o número máximo de clientes que pode processar ao mesmo tempo, terá também um endereço IP que lhe foi atribuído. No começo do seu runtime começa por criar o socket TCP por onde recebe comunicações da proxy, contendo o tipo de conexão. Se a conexão for do tipo UDP são usadas funções diferentes que permitem a ligação à proxy da maneira escolhida.

Ao enviar o comando DOWNLOAD o cliente pode escolher qual o tipo de ligação para a transferência do ficheiro, além de se quer ou não qualquer tipo de encriptação. Ao enviar um comando deste tipo, o servidor vai abrir o ficheiro como binário e envia o tamanho do ficheiro primeiramente. Ao receber o tamanho do ficheiro e como o buffer tem tamanho limitado, o cliente vai saber quantas vezes e qual o tamanho a ler do buffer até ler o ficheiro por completo.

Sendo assim a transferência do ficheiro é sequencial, transmitindo otimamente uma quantidade de bytes do tamanho do buffer. O ficheiro é escrito como binário também no diretório de Download do cliente. Esta abordagem permite transmitir quaisquer tipos de ficheiros, como exemplo texto e imagem.

A encriptação trabalha sobre a sequencia de bytes lida do ficheiro e a transmitir. Será depois desencriptada pelo cliente.

A proxy tem a função de guardar os ficheiros que por ela passam, para isso apenas é necessário abrir também um ficheiro binário para a qual são guardados os dados e estes são à mesma transmitidos para o cliente. No caso em que os dados são encriptados é expectável que os dados estejam na íntegra no ficheiro, mas como o proxy não tem acesso à chave de descriptação não será possível extrair qualquer informação destes.

Explicação de algumas funções:

Foi necessária a criação de funções (send_int e receive_int) para o envio e receção de inteiros nos diferentes protocolos, usadas no envio do tamanho do ficheiro.

Também a escrita e leitura dos ficheiros binários e a sua transmissão é feita por meio de funções (enviaStringBytesTCP e handleUDP), que procuram o ficheiro no diretório, o abrem e enviam consoante o protocolo. O tamanho a enviar é calculado com base nos tamanhos do buffer e do ficheiro em si.

Na proxy existe uma variável em memória partilhada (save) que dita se os ficheiros que passam são ou não guardados, dependendo do seu valor.