Disciplina: Teleinformática e Redes 2
Professor: João Gondim
Victor Landim 15/0023031
Lincoln Abreu Barbosa 14/0045023

Trabalho Final

Proxy HTTP

I. Introdução Teórica

Para o pleno entendimento do trabalho a ser apresentado,faz-se necessária a exposição de alguns conceitos. O protocolo TCP - *transmission control protocol* - é um dos mais importantes protocolos de comunicação da internet como a conhecemos hoje. TCP, que opera na camada de transportes, é um protocolo orientado a conexão que permite que dois *hosts* conectem-se e troquem informações em um canal bidirecional (aquele em que os dois *hosts* participantes podem enviar e receber pacotes de dados). Além disso, o TCP, em contraste com o protocolo UDP, tem o papel fundamental de oferecer um serviço com entrega confiável e livre de erros.

Neste contexto, temos o protocolo HTTP - *hypertext transfer protocol* - que é o protocolo da camada de aplicação que opera acima do protocolo TCP. HTTP é a base da internet, é um protocolo cliente-servidor em que clientes (comumente navegadores web) fazem requisições, chamadas de *requests,* a servidores e recebem como resposta, *responses,* recursos como páginas HTML ou arquivos estáticos como arquivos CSS e JS. Seu principal propósito é ser um protocolo simples e legível para humanos. Ainda, é dito que o protocolo HTTP é *stateless*, isto é, não existe dependência ou ordem entre requisições. Contudo, com a introdução de *cookies*, campos adicionais em um requisição podem indicar seu contexto. Dessa forma, mantém-se a filosofia do protocolo e fornece-se aos clientes coerência no serviço.

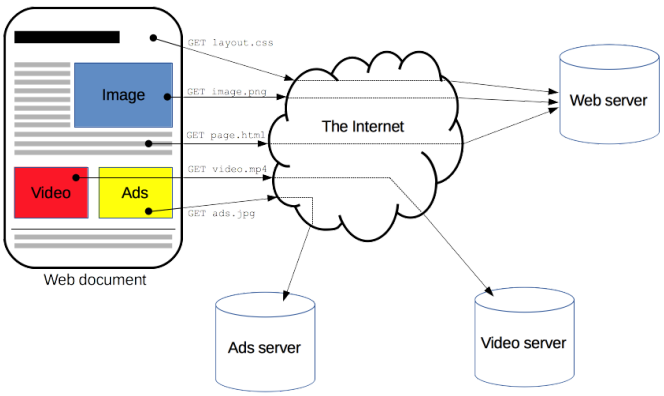

Figura 01 - Funcionamento do protocolo HTTP.

II. Funcionalidades do Sistema

A. Introdução

A proposta do projeto era que se fosse desenvolvida uma aplicação gráfica capaz de interceptar e editar requisições HTTP enviadas a partir de um browser. Além disso, deveriam ser disponibilizadas as funcionalidades de geração da árvore hipertextual de um determinado endereço web e, ainda, poder armazenar em disco todo um website.

B. Proxy

A interface da aplicação foi desenvolvida com a framework QT. Nela, pode-se criar o aspecto gráfico com facilidade por meio de um menu drag and drop. Ainda, QT disponibiliza um sistema chamado de signals and slots, que é um espécie de implementação do padrão observador, que permite que diferentes partes e camadas da aplicação fiquem cientes de certos comportamentos, o que permite que tomem ações necessárias. Segue alguns exemplos de tela:

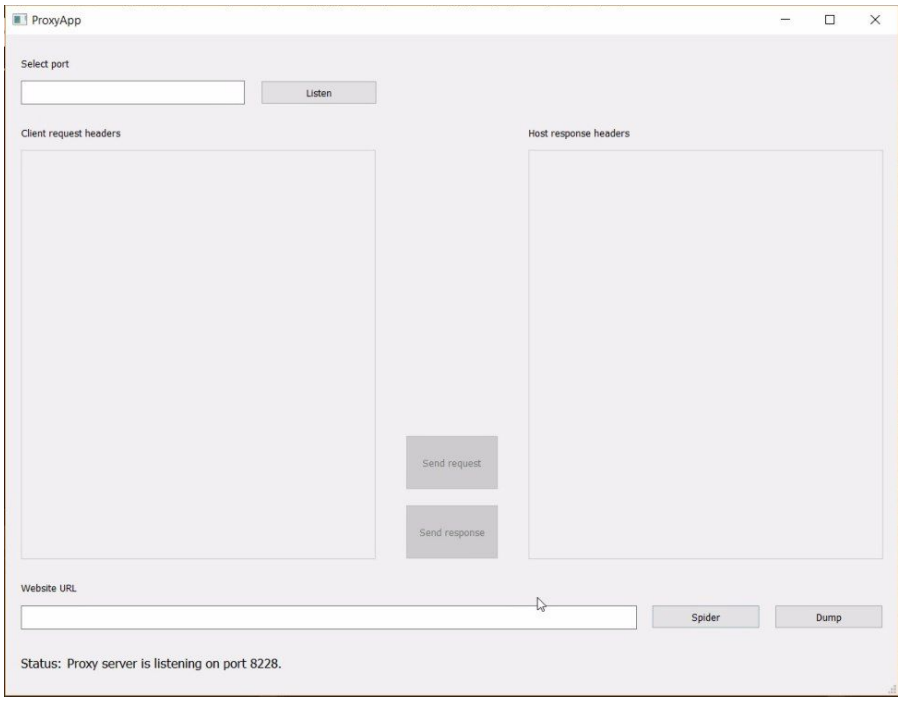

Figura 02 - Interface principal, aguardando requisições.

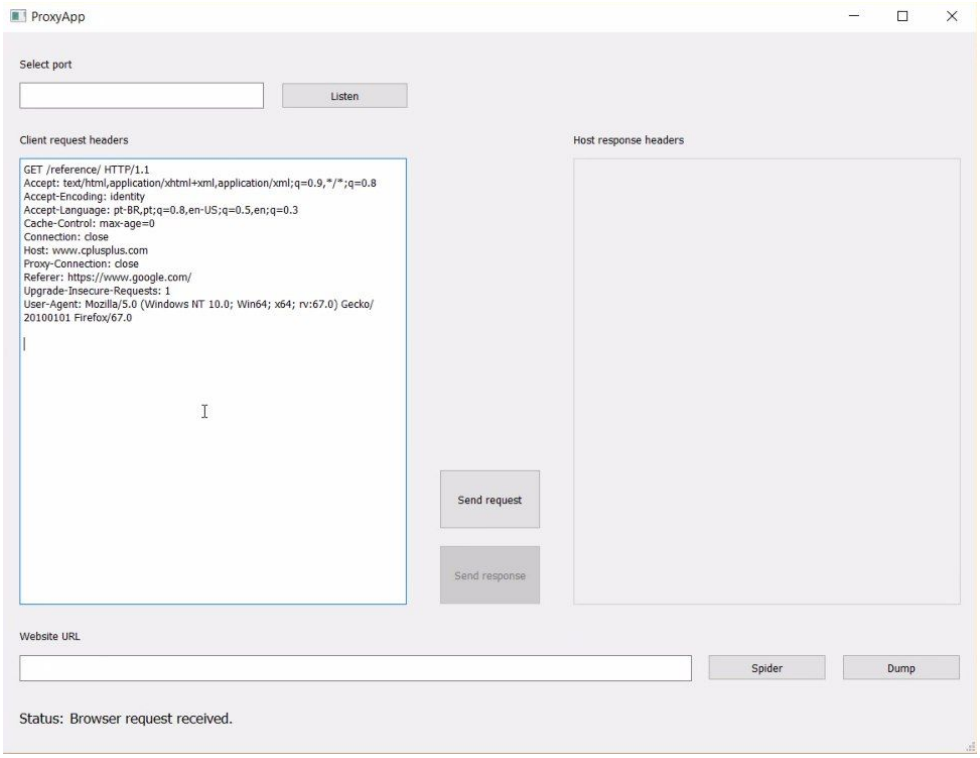

Figura 03 - Request do browser recebida.

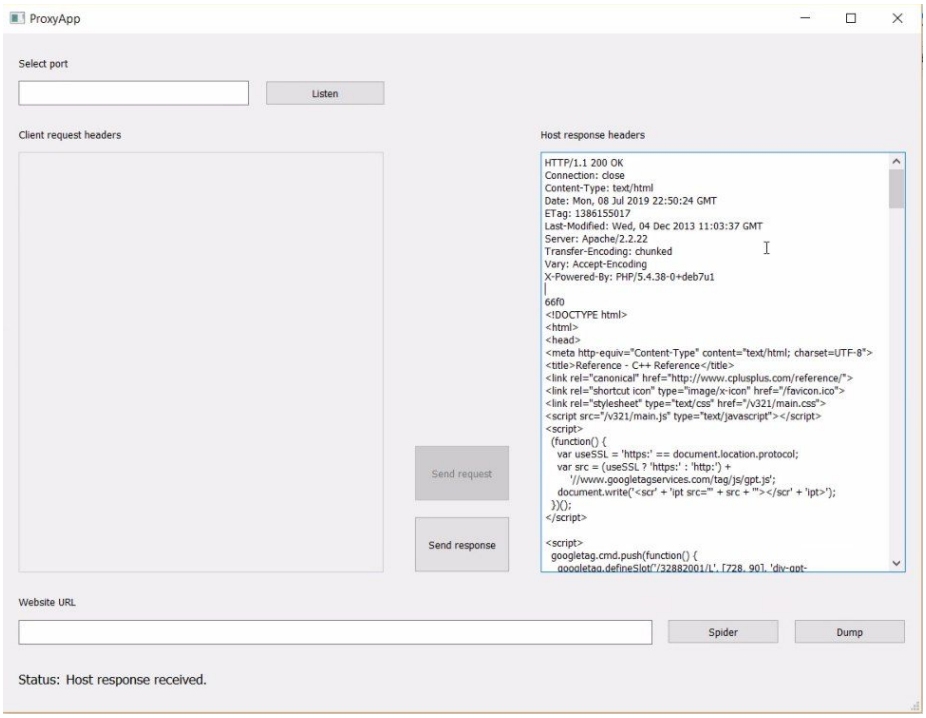

Figura 04 - Response do servidor recebida.

De maneira geral, o sistema funciona da seguinte forma: primeiro configura-se o browser para que suas requisições sejam encaminhadas para um proxy server em uma determinada porta. Em seguida, solicita-se uma página HTML, por exemplo, http://cplusplus.com. Então, a request é capturada pela aplicação e deixa o browser aguardando. Então, o usuário pode inspecionar o conteúdo do request e, se desejar, editá-lo. Em seguida pode-se pressionar o botão “Send request”, que envia a requisição e recebe os dados do servidor de destino do request. Então, novamente, a resposta é apresentada para o usuário, que pode editar o conteúdo antes de finalmente encaminhar a resposta final para o browser. Assim que a resposta é recebida pelo browser, a aplicação já se coloca no estado inicial e volta a escutar por conexões, repetindo o ciclo. O usuário, ainda, pelo botão listen, pode alterar a porta que o proxy escuta. A nível de implementação, detalhes importantes podem ser reparados. No processo de escutar, receber dados do browser e do servidor de destino, esta aplicação configura e abre sockets para permitir a comunicação, ou seja, uma implementação de baixo nível em uma aplicação de redes. Além disso, a aplicação cuida de detalhes importantes, como realizar o parsing de requests e responses, extraindo e armazenando headers, status e body.

C. Spider

O Spider funciona percorrendo em os links -URLs- encontrados ao fazer o parsing do HTML (possuem marcador href) até uma profundidade máxima, desconsiderando páginas já analisadas e retornando uma lista ordenada de todos os links encontrados com suas respectivas profundidades.
O resultado do Spider é então armazenado em arquivo com os símbolos “|->” para auxiliar na compreensão da árvore de referências.

D. Dump

O Dump funciona de forma análoga ao Spider, no entanto salva cada html analisado em um arquivo e simula a estrutura de profundidades através de pastas e subpastas, alterando as referências “href” de URLs no HTML, sejam relativos ou absolutos, pelos endereços absolutos dos arquivos e pastas criadas no disco local.