Inspetor HTTP baseado em Proxy Server
Teleinformática e Redes 2 - 1/2019

Danilo José Bispo Galvão 12/0114852
Guilherme de Oliveira Silva 11/0012496

Introdução Teórica

Um servidor proxy se comporta como um interceptador de *requests* e *responses* HTTP de clientes. Ao realizar uma requisição, o cliente se conecta ao servidor proxy, que, por sua vez, intercepta a requisição. Se o objeto requisitado estiver guardado em *cache*, o servidor proxy não se conecta ao servidor e já retorna o objeto requisitado ao cliente, diminuindo a carga de requisições no servidor, caso contrário, a requisição é então repassada para o servidor na web que retorna o objeto requisitado.

Um servidor proxy pode ter várias funções, como filtragem de conteúdo(sites inadequados ou não-confiáveis), monitoramento, controle de acesso e caching.

HTTP e TCP

O HTTP(HyperText Transfer Protocol) é responsável pela estrutura das mensagens que são trocadas na web. Cada mensagem é endereçada por uma URL que é dividida em nome do host e nome do caminho do objeto. Os navegadores são normalmente encarregados de implementar o cliente HTTP, que realiza as requisições com o endereço fornecido pelo usuário, o processo começa com uma conexão TCP, onde uma porta é determinada para a troca de mensagens, após essa definição, a requisição é realizada. Os métodos de requisição mais conhecidos são: GET, POST, PUT E DELETE, cada um deles com sua particularidade, na mensagem da requisição também são mandados a URL e a versão do HTTP. Na resposta temos alguns campos principais:

- Versão do HTTP;
- Código do status(Status code);
- Data(Date);
- Servidor(Server);
- Última vez modificado(Last-modified);
- Tamanho do conteúdo(Content-Length);
- Tipo do conteúdo(Content-Type);
- Conexão(Connection): Closed;

O TCP é o protocolo da camada de transporte orientado a conexões que determina os sockets onde os datagramas serão enviados, como a camada de aplicação pode definir qual dos dois protocolos ela irá utilizar(UDP ou TCP), fica a cargo da aplicação desejada definir qual será o protocolo utilizado, no nosso caso, o HTTP escolhe o TCP por sua característica de transferência confiável e ordenada, além de ter um controle de congestionamento.

Spider e Cliente Recursivo

Spider é o nome dado a um mecanismo que rastreia páginas e identifica todos os hyperlinks que ela possui, gerando uma árvore hipertextual de profundidade customizável que representa todas as URLs dentro do domínio da página raiz.

O cliente recursivo tem como objetivo realizar o *dump* da página, isso é, baixar todos os objetos disponibilizados pela página e as páginas subsequentes da árvore, incluindo o seu texto HTML, mantendo a mesma hierarquia da árvore hipertextual em um diretório de pastas gerado no diretório de usuário após o download.

Arquitetura

O sistema está distribuído entre arquivos .cpp e .hpp de modo a organizar as funções e declarações de variaveis globais. Foram criadas seis classes para a resolução dos problemas propostos, as classes são:

- HTML Parser
- HTTP Request
- HTTP Response
- Proxy Server
- Spider
- String Functions

A função main é responsável por receber a porta fornecida pelo usuário, chamar a função de interface gráfica e executar o sistema. As seções a seguir irão se aprofundar no funcionamento de cada classe.

HTML Parser

A classe de parser é responsável por extrair informações do corpo da mensagem, possuindo quatro funções:

- getUrl;
- getSource;
- getImport;
- getHtml;

HTTP Request

A classe HTTP Request é responsável por tratar os requests inspecionados a partir do servidor proxy. Essa classe possui cinco atributos: método, url, versão, campos e corpo. Esses atributos juntos formam a mensagem de request. Os métodos dessa classe são:

- print;
- tratarConexao;
- avaliaMetodo;
- montaRequest;

O método print é responsável por mostrar na tela os atributos da classe. O método de avaliação testa se o método do request é GET ou não, sua resposta é uma variável booleana. O método de tratamento é responsável por colocar no campo connection da mensagem, o valor close e no campo accept encoding o valor identity. Por fim, o último método é responsável por montar o request.

HTTP Response

A classe de response é semelhante ao request no sentido de montar a mensagem, porém neste caso a mensagem de response. A classe possui quatro atributos:

1. codigoStatus;
2. versao;
3. campos;
4. dados

E seus métodos são:

- print;
- montaResponse;

A função de print semelhante a classe anterior, mostra os atributos para o usuário. A função de montar o response monta uma lista com os atributos para criar a mensagem de response.

Proxy Server

Essa classe é responsável por começar a execução do sistema, seus métodos estão listados:

- init
- get client request
- reply client
- make request

A função de init realiza a conexão a partir da porta fornecia, ou da porta 8228 por convenção, cria o socket do servidor e realiza o bind. A função get client request monta um request do cliente. O método de reply client testa se o envio da resposta do cliente no socket deu certo e fecha o socket do cliente. Por fim o método make request realiza de fato o request com o endereço do servidor e retorna um reply.

Funcionamento

A interface gráfica possui uma série de botões que ajudarão o usuário a usar o sistema. Primeiramente ao clicar no botão de start o inspetor HTTP é iniciado pegando as requisições e respostas do browser, porém só é mostrado na tela as mensagens se o usuário clicar nos botões de request e reply. Abaixo das caixas de texto que mostram as requisições e respostas possui uma caixa de texto que é responsável por receber a URL que o usuário deseja realizar o spider, ao clicar no botão go ele gera a árvore. Existe também uma caixa de texto onde o usuário deverá colocar uma URL que ele deseja realizar o dump de todo o conteúdo da página web. Lembrando que as páginas deverão estar no protocolo HTTP.