Relatório Trabalho Teleinformática e redes 2

PlzDontProxy

Alunos:
Luisa Sinzker Fantin 14/0151893
João Paulo Tavares 13/0029335

Apresentação teórica

- Protocolo HTTP

O protocolo de transferência de hipertexto (HTTP) veio com o objetivo de facilitar a vida dos usuários da recém-nascida world wide web, responsável pelo WWW nos endereços dos sites por toda a web. Esses sites usam o hipertexto para facilitar a navegação dos usuários, fazendo com que os elementos desses sites sejam mais do que um texto puro e simples, como num bloco de notas, podendo muitas vezes serem clicáveis e redirecionados para outras páginas com melhor explicação, um som, imagem ou conteúdo relacionado.

O HTTP baseia-se no princípio de clientes e servidores, onde um navegador da web é um cliente que faz requisições para um servidor, buscando dados da página ou funcionalidades de interesse do cliente. O protocolo HTTP intermedia essa conexão entre o cliente e o servidor de diferentes formas a fim de melhorar esse processo tanto para o cliente quanto para o servidor. No caso do cliente, esse protocolo busca melhorar o tempo de resposta para carregamento de páginas e também a experiência geral do uso de sites na web. Para o servidor, no caso de um com grande número de acessos, o HTTP pode melhorar na utilização de um servidor de cache, assim evitando o carregamento desnecessário de alguns itens múltiplas vezes, melhorando o acesso dos seus usuários e economizando banda do servidor.

Esse protocolo não é algo novo, o HTTP é utilizado desde 1990 na web e suas versões são formalizadas através de RFCs (Request for comment), a versão mais recente é de 2015 (HTTP 2) – HTTP 3 deveria ter saído em 2018 – e a sua versão mais antiga é de 1991. Com essas RFCs o HTTP evoluiu de várias formas, ao longo dos anos novos métodos foram adicionados e novas funcionalidades vieram junto, tanto para melhorar a experiência do usuário final como também a segurança.

Existem oito métodos que o HTTP define para se fazer a comunicação entre clientes e servidores, esse métodos evoluíram com o passar das versões HTTP por meio das RFCs, assim para que haja a comunicação entre um servidor e cliente não é necessário que todos os métodos sejam implementados, por exemplo um servidor só precisa implementar dois (GET e HEAD) dos oito métodos definidos.

Um navegador utiliza esses métodos para fazer uma requisição à um servidor, por meio de uma solicitação HTTP afim de mostrar uma página da web, para isso também manda com a solicitação vários dados do cliente, como qual o navegador, idioma, entre outros, assim a resposta a essa solicitação será a mais adequada ao usuário e mostrará melhores informações

- Protocolo TCP

O protocolo de controle de transmissão é um dos mais importantes protocolos que foi definido para que tivéssemos a internet como ela é hoje, é por meio dele que diversas aplicações funcionam e garantem o oferecimento dos seus serviços da forma como é feito. Esse protocolo define como deve funcionar o envio de pacotes pela rede, assim ele foi definido com características técnicas importantes para que houvesse o bom funcionamento de uma rede tão grande como a internet.

O TCP é um protocolo da camada de transporte, provendo vários serviços importantes para a camada de aplicação, onde estão vários outros protocolos, um deles o HTTP, além do FTP e SSH que são protocolos igualmente importantes para o bom funcionamento da internet e que se beneficiam fortemente das características do TCP.

O TCP trabalha em conjunto com o IP, por isso em vários lugares essa associação é feita quase sempre e temos a sigla TCP/IP. O IP é um protocolo de camada de rede e o TCP o utiliza como complemento.

Dentre os diversos benefícios oferecidos pelo TCP a grande evolução se deu por conta da entrega confiável de pacotes de dados, da verificação de erros e da entrega sequencial desses pacotes, dessa forma uma aplicação pode ser implementada sem precisar se preocupar com esses pontos na hora de se comunicar com outros computadores.

Há também definições de como o protocolo TCP deve trabalhar para fazer suas conexões, afim de evitar perda de dados, o TCP usa o método conhecido como 3 way handshake em suas conexões, um método que primeiro checa se uma conexão com um determinado host é possível ser realizada, espera uma resposta desse host e manda uma terceira reposta dizendo para que seja efetuada a conexão antes que os dados requeridos sejam de fato enviados.

Todos esses procedimentos adotados pelo protocolo TCP vem com o custo de maior latência na comunicação, mas na maioria das vezes essa latência não é tão importante quanto as garantias oferecidas por esse protocolo. Em algumas aplicações essa garantia não é tão importante e a velocidade acaba sendo crucial, mesmo ao custo de perda de alguns pacotes de dados, assim essas aplicações usam outro protocolo, chamado UDP, que é um protocolo mais simples e que não faz verificações igual ao TCP em busca de reduzir a latência na hora da comunicação.

- Proxy Web

Um proxy é um intermediário entre conexões feitas por hosts, normalmente entre um computador e um servidor, com o proxy (que também é um servidor, de proxy nesse caso) no meio dessa ligação. Costumeiramente um proxy é utilizado para que um computador, ou vários, numa rede interna tenha acesso a conexão com a internet por meio de outro computador, agindo nesse caso como um servidor proxy.

Existem inúmeros tipos de servidores proxy, cada um com uma função específica, os mais comuns hoje são servidores proxy do tipo web, aqueles que de alguma forma estão vinculados a comunicação de um cliente com um servidor na internet. Dentre os servidores proxy desse tipo ainda existem outros mais específicos.

Um proxy do tipo web funciona intermediando a comunicação entre um cliente e um servidor, onde um cliente manda a requisição ao servidor, o proxy recebe essa requisição e pode ou não

alterá-la antes de enviar ao servidor, após o envio o servidor responde a requisição, o proxy recebe essa resposta e também, novamente, pode ou não alterá-la antes de enviar de volta ao cliente que realmente fez a requisição.

No exemplo de um proxy intermediando uma conexão entre um cliente e um servidor, pode ser vantajoso para o cliente um proxy que garante o seu anonimato, assim as pessoas por trás do servidor não obtêm as informações sobre o cliente que está acessando o seu conteúdo. Enquanto alguns usuários podem achar isso vantajoso outros podem não se importar em divulgarem suas informações.

Outro tipo bem comum de proxy server é um proxy cache server, responsável por ter uma lista de arquivos mais acessados e buscados em seu servidor, assim evitando que algum cliente precise buscar externamente algo que ele já tenha, economizando banda e tempo. Quando esse servidor não possuir o conteúdo buscado esse proxy faz a requisição do conteúdo, passa ao cliente e também armazena em seu cache.

Motivos para utilização de um proxy server não faltam, muitos alegam falta de privacidade e recorrem a servidores proxy para dificultar o rastreamento e o compartilhamento de suas informações na internet. Empresas também usam esses servidores para poder saber o que está sendo acessado em sua rede corporativa, bloquear conteúdos não permitidos e ter um controle maior de sua banda larga.

- Spider

Um Spider no contexto desse trabalho é um rastreador de URLs que, dado um site, ele busca todos os links que levam a outros sites do mesmo domínio dentro desse site. Fazendo isso para todos os links encontrados obtém-se uma estrutura onde o site principal leva a várias páginas filhas, normalmente com conteúdo relacionado ao principal.

Num contexto mais amplo, um spider possui diversas aplicações e pode ser utilizado para várias finalidades, comumente é usado para análise de sites e de metadados nesse site, assim como descobrir links quebrados e conteúdo duplicado.

Para implementar um spider um site precisa ser varrido buscando-se todos os links contidos nele e analisando esses links até que se chegue a uma página que não contenha mais links, muitos programas se propõe a fazer esse tipo de coisa, existem web-crawlers próprios para essa função e diversos programas pagos que fazem isso num nível profissional.

- Cliente recursivo

O Spider pode ser utilizado para criar uma árvore com todos os links obtidos por ele, com isso temos uma árvore que pode ser consultada e que tenha todos os links de um domínio, dado um site principal. O cliente recursivo faz um dump do conteúdo de uma dada URL, obtendo assim todo o conteúdo dessa URL e no contexto desse trabalho modifica-se as referências para que esse conteúdo possa ser visualizado no navegador e identificado por esse como local.

Esse cliente recursivo pode ser usado para que se saiba quais objetos um site está disponibilizando para download e o que é contido por cada um desses objetos, assim em uma análise seria possível identificar problemas ou erros associados ao processo de download desses objetos.

Arquitetura

- HTTP Request

Analisa as informações contidas nos campos do HTTP, como cabeçalho, para que possa ser realizado manipulação nesses dados caso seja necessário. Pode ser necessário selecionar alguns parâmetros dentro desse cabeçalho e descartar outros.

Também é responsável por transformar o texto da requisição HTTP para um tipo que o resto do programa possa entender e também fazer o inverso, transformar um texto do programa num tipo que o HTTP possa compreender (ASCII).

- NetSocket

O NetSocket cria uma conexão com o servidor, usando TCP, (caso o servidor exista), responsável pela comunicação com o servidor.

Cria uma conexão com o IP informado no objeto httprequest criado, após traduzir o DNS no IP, envia a requisição ao servidor e quando chega a resposta do servidor faz o processo inverso.

Na criação do objeto, quando recebe a resposta do servidor pode acontecer de não chegar todos os dados juntos, precisando assim realizar a correta organização desses dados contidos na resposta da requisição.

- AppSocket

O AppSocket cuida da parte de conexão com o browser, esse módulo é responsável pela conexão com o navegador e por manter essa conexão ativa, utilizando o protocolo TCP.

Cria objetos httprequest à medida que o navegador manda requisições para o proxy e quando as recebe de volta manda novamente ao navegador.

Fica monitorando a porta informada pelo usuário no início do programa, por padrão a porta é 8228, mas outra pode ser informada, apenas precisa ser corretamente configurado no navegador para que não haja problemas.

- Spider

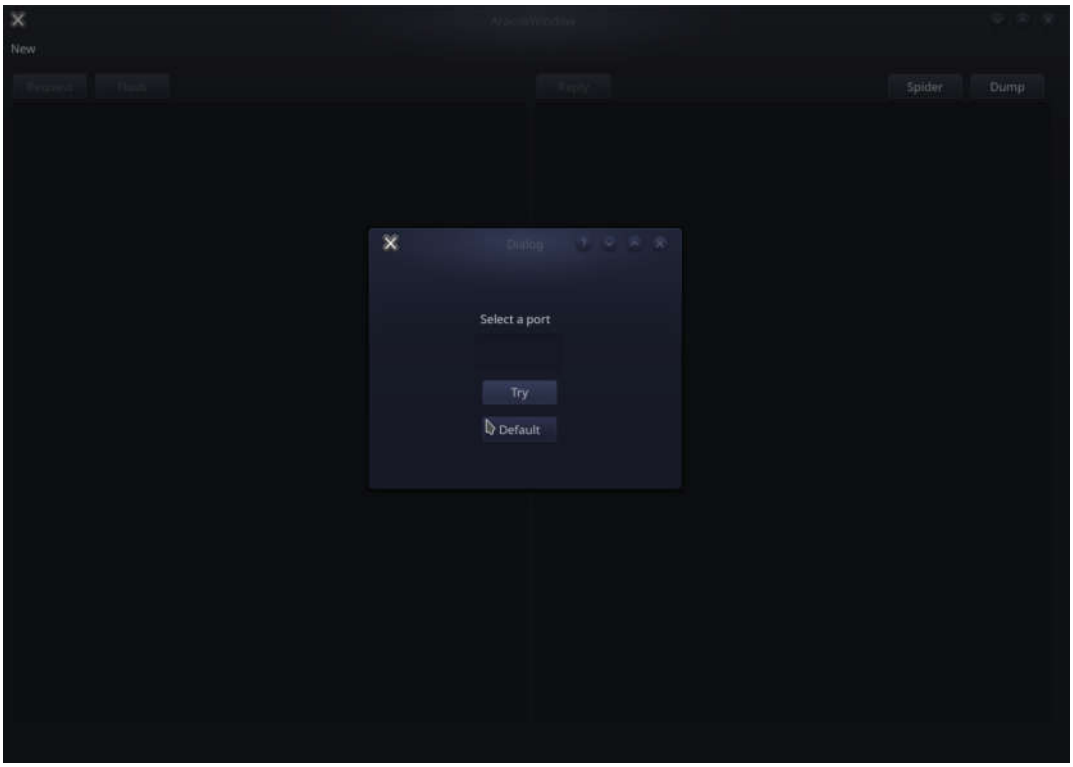
O Spider é a parte que busca todos os links dentro de uma URL, varrendo todo o código HTML para isso. Procura por campos específicos que indicam a presença de outra URL dentro daquela página, assim aplicando o mesmo procedimento para essa URL até que seja encontrada uma página sem novos links, imprimindo tudo na tela, montando a árvore hipertextual desejada na especificação do trabalho.

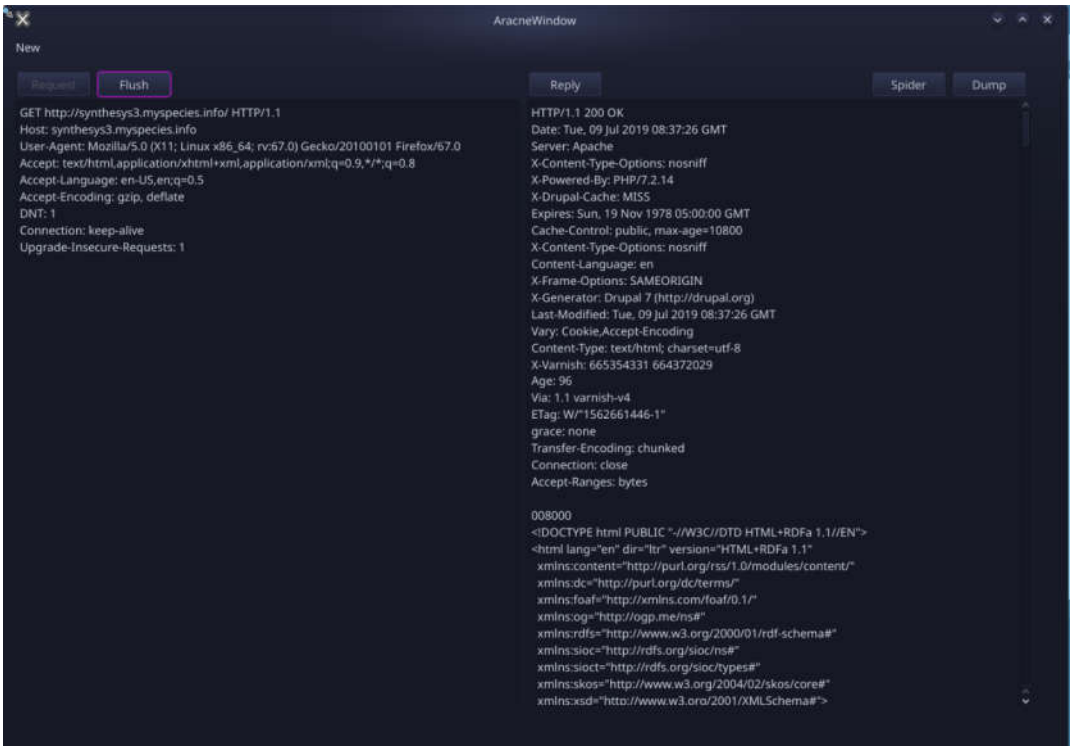
- Cliente recursivo

Parte responsável por baixar todo o conteúdo de uma dada URL, aproveita das funcionalidades implementadas pelo spider para obter URLs mais facilmente e obter o conteúdo dessas páginas desejadas. O dump desse conteúdo é mostrado na tela, todos os objetos baixados de uma dada URL, além da modificação das referências para apresentação no navegador como conteúdo local.

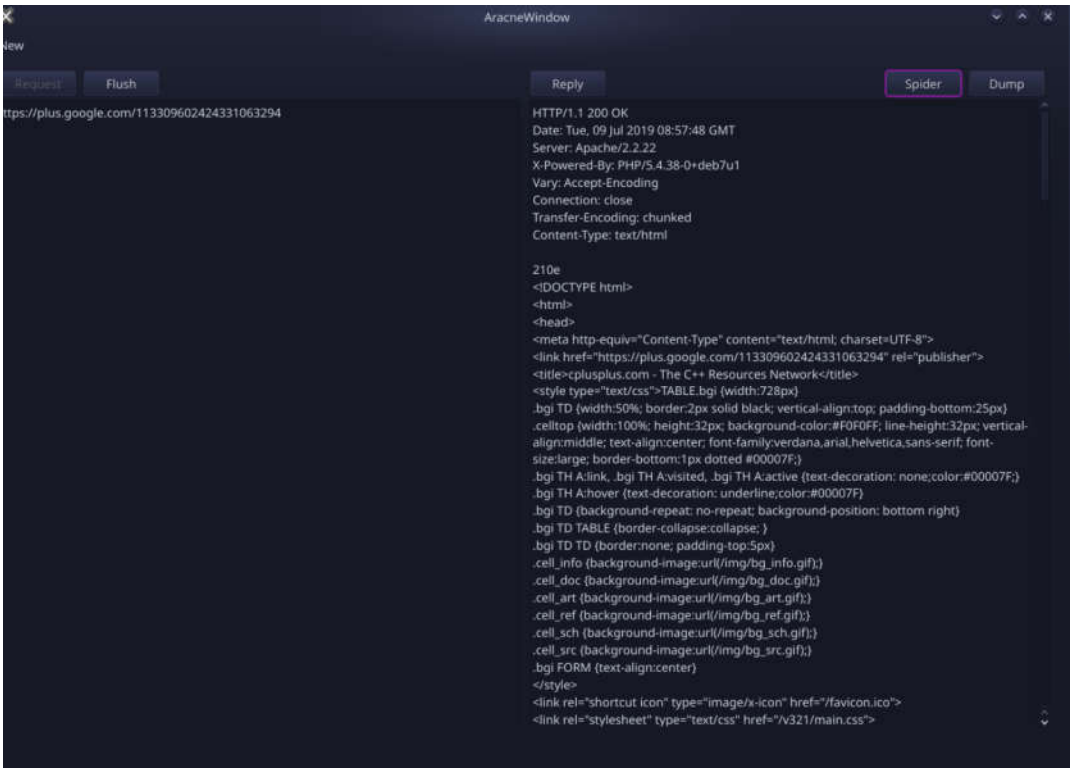
Screenshots

Proxy – requisição e resposta





Spider – URLs na página e árvore



Cliente recursivo – Dump do conteúdo