Projeto Final de Teleinformática e Redes 2
Implementação de uma aplicação de Proxy Server

Nome: Amanda Aline Figueiredo Carvalho - 15/0115997
Nome: Bruno Justino Garcia Praciano - 13/0006912
Nome: Bruno Viana de Siqueira - 14/0133151

04 de Dezembro de 2017

1 Resumo

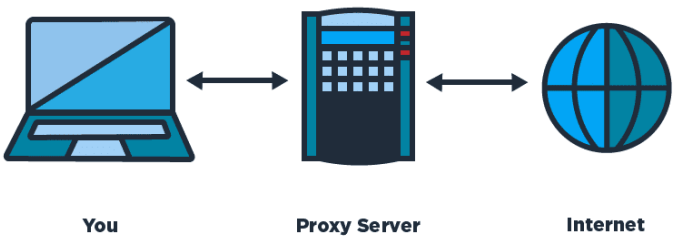
Esse trabalho tratará sobre o Proxy Server Web, TCP e o protocolo HTTP, bem como da aplicação da teoria aplicada em sala de aula.

2 Apresentação Teórica

2.1 Proxy Server Web

No contexto de redes de computadores, um servidor proxy é um servidor, seja ele um sistema de computador ou mesmo uma aplicação, que atua como intermediário para pedidos de clientes que buscam recursos de outros servidores. Um cliente se conecta ao servidor proxy, solicitando algum serviço, como um arquivo, conexão, página da Web ou outro recurso disponível de um servidor diferente e no momento em que o endereço de um site é digitado no navegador, uma solicitação é enviada ao proxy, que então realiza esta solicitação ao servidor no qual o site é hospedado, devolvendo para o cliente o resultado.

O proxy foi inventado para adicionar estrutura e encapsulamento a sistemas distribuídos. Hoje, a maioria dos proxy são web, e entre suas características estão o fácil acesso ao conteúdo da internet, proporcionar anonimato durante a navegação e possibilidade de ignorar o bloqueio de endereço IP. O aumento constante do uso de serviços web, como o acesso a sites, sistemas e diversos outros aplicativos que operam online, podem gerar contratempos ao mesmo tempo que trazem comodidade e agilidade, como questões de segurança. A utilização de servidores proxy surgiu como opção de ferramenta de segurança e controle, inclusive de acesso a páginas indesejadas.



2.2 Protocolo HTTP

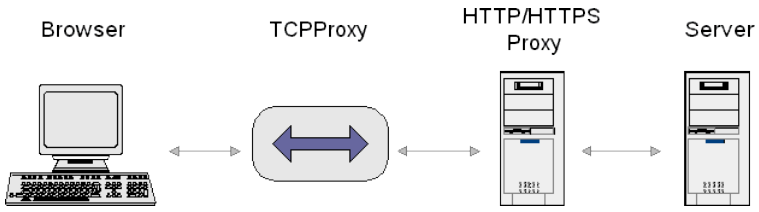
O servidor proxy faz o intermédio da conexão entre o dispositivo e o cliente, e por conta disso, tem controle total sobre o tráfego em si, podendo permitir ou bloquear de acordo com a arquitetura construída, conforme será descrito no desenvolvimento. Muitas configurações podem ser atribuídas ao proxy, fazendo com que ele permita e restrinja URLs de acordo com o interesse. O proxy web

permite controlar os serviços acessados na internet através do protocolo HTTP, sendo responsável pela gestão de acesso a sites e outras aplicações baseadas no protocolo.

Em suas requisições, o cliente envia um cabeçalho que faz a identificação dos serviços utilizados e outras informações.

```
HTTP/1.1 200 OK
Date: Mon, 31 Mar 2014 16:55:30 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.6.29.7-3566
Cache-Control: max-age=0, no-cache, must-revalidate
Vary: Accept-Encoding,Cookie
X-Node: askapacherackweb0
X-UA-Compatible: IE=Edge,chrome=1
Content-Length: 55069
Keep-Alive: timeout=4, max=46
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Content-Language: en
```

O Web Proxy Caching atua no armazenamento local de páginas da internet e arquivos disponíveis em servidores remotos, tendo em vista que busca objetos primeiramente no sistema de armazenamento local, seja a memória ou disco. Assim, ao receber a solicitação, o servidor acessa o conteúdo e o armazena para utilização futura. Em um próximo acesso, o proxy responde com o conteúdo armazenado, aumentando a velocidade de resposta, uma vez que não se consume banda da internet. Ao implementar um proxy cache é possível salvar em diretórios as URLs já carregadas pelo menos uma vez, mas não por tempo indefinido, pois age de acordo com a capacidade de armazenamento da máquina. Abaixo está o exemplo de funcionamento de um proxy HTTP

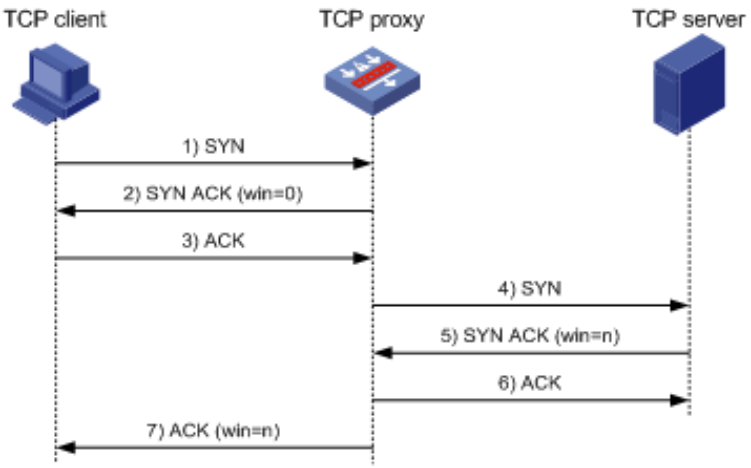


É uma característica do HTTP utilizar os serviços disponibilizados pelo TCP, pois geralmente os servidores HTTP usam a porta 80 do TCP. O HTTP/1.0 é diferente do HTTP/1.1, pois este último utiliza conexões TCP persistentes. Graças ao uso de um proxy, é possível filtrar as conexões à Internet analisando, tanto os pedidos do cliente, quanto as respostas do servidor.

2.3 TCP

O TCP é um protocolo da camada de transporte do Modelo OSI e é atua na maioria das aplicações da internet, como o SSH, FTP, HTTP, portanto, a World Wide Web. O Protocolo de controle de transmissão provê confiabilidade, entrega na sequência correta e verificação de erros em pacotes de dados, entre os diferentes nós da rede, para a camada de aplicação.

Ocorre troca de pacotes entre cliente e servidor. Assim, são estabelecidos parâmetros para a conexão. Este fenômeno ocorre por meio do handshake. O handshake começa quando o cliente envia um segmento de sincronização que contém um número inicial de sequência, ao servidor. O servidor, ao receber o segmento de sincronização, envia uma mensagem de aceite da conexão (ACK). Em complemento, é enviado o número inicial de sequência do servidor. Por fim, o cliente confirma o recebimento do aceite enviando um segmento de ACK.



3 Desenvolvimento

3.1 Arquitetura do sistema desenvolvido

O sistema proxy foi desenvolvido na linguagem C. O proxy web caching está atuando conforme o esperado, assim como a filtragem de termos. O arquivo se chama proxy cache.c e contém todas as funções utilizadas no trabalho. Os requisitos do trabalho se encontram nas funções desse arquivo C. A pasta contém:

- Um arquivo C, com todas as funções implementadas documentadas.
- Um README contendo as informações de compilação e execução
- Diretório contendo os sites carregados em cache.
- Arquivos txt para filtragem dos domínios permitidos e proibidos, além de termos não permitidos.

3.2 Documentação do código

A documentação das funções está no próprio arquivo C, com uma descrição breve do funcionamento de cada função.

3.3 Implementação do Funcionamento Básico

A arquitetura do proxy foi pensada da seguinte maneira: o proxy recebe uma solicitação de um host, busca as informações necessárias na requisição, como cabeçalho, path e IP. Após, o cabeçalho do início da requisição passará a ser o proxy no host de origem. Associado a isso, há também o cache, que será abordado nas subseções a seguir.

O código foi fatiado em funções num mesmo arquivo, porém cada uma delas possui uma participação específica na execução do proxy.

- main - função principal, que chama os argumentos principais de funcionamento básico.
- verifyWhiteAndBlackList - Retorna um inteiro indicando se está na blacklist (-1), se está na whitelist(1), ou em nenhuma outra lista (0). O Buffer da dados será comparado ao arquivo de blacklst e whitelist.
- verifyDenyTerm - Retorna um inteiro indicando se o buffer contém um deny term. O Buffer da dados será comparado ao arquivo de deny terms.

- *get in addr - Função de passagem de endereços na socket.
- directory - Essa função cria os diretórios para o proxy cache. Optou-se por fazer dessa forma ao invés de salvar apenas as URLs em txt para efeitos de controle se a página já foi ou não acessada e se está gravada em memória, e verificação dos cabeçalhos. A saída fornece uma flag se a página está em cache (já foi acessada) ou não (primeira vez de acesso).
- connect host - Responsável por fazer a conexão com o host, setando seus parâmetros.
- cache - Carrega o buffer de armazenamento temporariamente para o cache.
- request - trabalha nas requisições de servidor, armazenamento em buffer pro cache de acordo com os requisitos de rede e filtragem.
- start server - Responsável por iniciar o socket. Retorna o inteiro retornado pela função accept. sockfd será iniciado pela função.
- send file - Passagem de arquivos para montagem de novo cabeçalho.
- set server - Configura o servidor chamanado as funções declaradas anteriormente, bem como algumas da biblioteca socket.h. Utiliza como base Socket API.
- deniedLogWrite - Cria e escreve um arquivo de log com os denied terms. Denied term será escrito no arquivo de log.
- blackLogtWrite - Cria e escreve um arquivo de log com as url blacklist. Buffer contém a URL da blacklist.
- whiteLogWrite - Cria e escreve um arquivo de log com as url da white list. Buffer contém a URL da whitelist.

3.4 Filtragem das requisições

A implementação foi feita baseado na premissa de que se o proxy da página acessada estiver na Whitelist ou na Blacklist, terá que verificar um terceiro termo: o deny term. Caso o domínio não estiver entre nenhuma destas listas, Black ou White, é necessário checar se esse termo pode ser acessado ou não, em DenyTerms.txt. Logs são emitidos para acompanhar o funcionamento.

3.5 Caching

A cache é o ambiente de armazenamento de páginas HTTP que são usadas rotineiramente. Essa técnica serve para otimizar o tempo de acesso e consumo de banda larga. As funções utilizadas foram citadas anteriormente em "Funcionamento Básico".

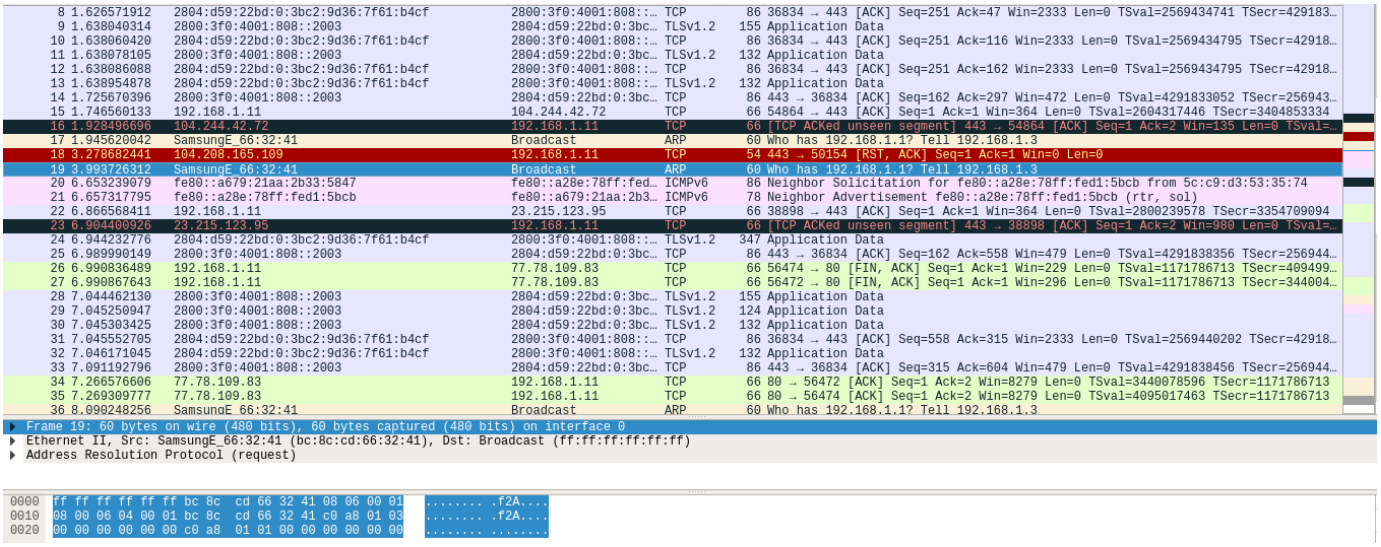
4 Conclusão

Esse trabalho nos permitiu aprofundar conhecimento sobre aplicações do que era visto apenas em sala de aula, de forma teórica. Foi possível ver o funcionamento dos protocolos e suas funções, como é o comportamento e importância de cada camada organizada. As maiores dificuldades enfrentadas estavam associadas à manipulação de algumas bibliotecas e desenvolver funções de filtragem. Por meio desse trabalho, foi possível colocar um proxy web para funcionar e entender melhor o funcionamento da Web, e protocolos HTTP e TCP.

5 Simulações

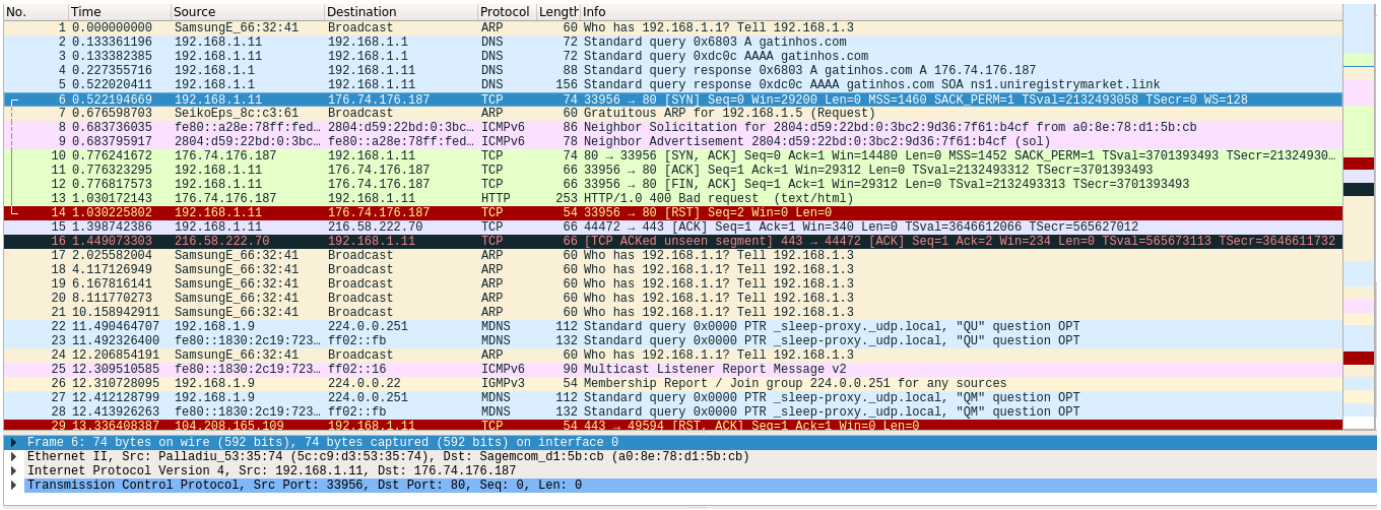
Foram feitas simulações utilizando o Wireshark e também capturas de tela a partir da execução do código em localhost. A seguir estão exemplos de simulações feitas e uma breve descrição de sua funcionalidade.

1) Simulação no Wireshark após acionarmos a Black List.



2) Simulação no terminal do Linux após acionarmos a Black List.

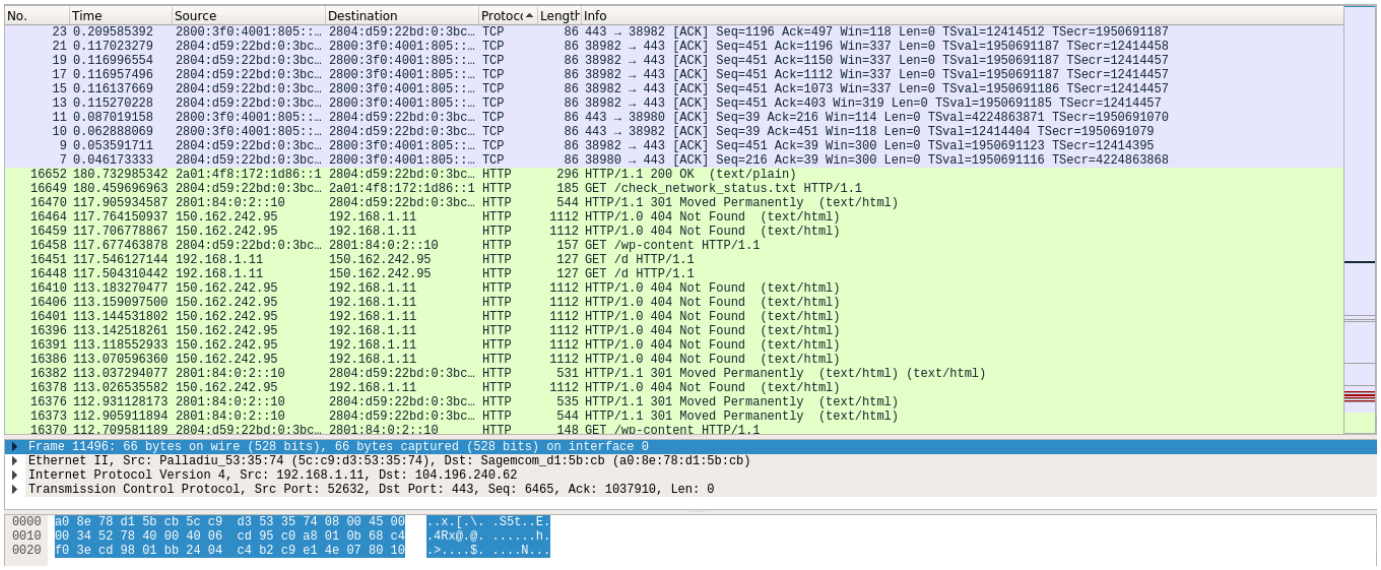
3) Simulação no Wireshark após utilizarmos Deny Terms.



4) Simulação no terminal do Linux após utilizarmos Deny Terms.

```
Connected client:
server:got connection from 127.0.0.1
host = gatinhos.com
path = (null)
Port = 80
..........
From Host
Deny Term found: gatinhos
```

5) Simulação no Wireshark após utilizarmos a White List.



6) Simulação no terminal do Linux após utilizarmos a White List.

```
Connected client:
server:got connection from 127.0.0.1
host = greattreasures.org
path = gt/index.html
Port = 80
..........
From Host
No deny terms found

GET /gt HTTP/1.1
Host: greattreasures.org
Connection: close
```