Relatório de desenvolvimento do Proxy

Helena Schubert Silva
10/0012311

Laís Mendes Gonçalves
11/0033647

***Resumo*—O proxy desenvolvido para a disciplina Teleinformática e Rede 2 foi um trabalho feito em C, usando sockets HTTP, e com outras funcionalidades além do proxy em si: uma inspeção de tráfego, com possibilidade de alteração das respostas, um spider, mostrando a relação de diretórios de um dado site, e um dump, possibilitando o download das páginas do site, e navegando por ele offline.**

I. Introdução

Proxy é um intermediador entre as requisições de um cliente (no caso, o navegador) e as respostas do servidor. Para o servidor externo, o proxy é visto como cliente, enquanto para o navegador, o proxy é visto como servidor, pois aquele é o que manda as respostas ao cliente.

Tendo esse papel de intermediador, o proxy é capaz de capturar e alterar as requisições, que consiste na funcionalidade de inspeção. E, recebendo as respostas, é possível abrir os pacotes e lê-los, procurando as as referências a outros objetos, que são usadas no spider e no dump.

Spider é uma funcionalidade que mostra as pastas e os caminhos dos objetos de um servidor, relacionando os links de um domínio entre si. Já o dump utiliza a estrutura do spider para gerar pastas homônimas na máquina do cliente e salvar os objetos recebidos em suas respectivas pastas, abrindo cada pasta de modo recursivo, aos moldes do programa *wget*.

II. Conceitos Teóricos

A. Proxy Server Web

O proxy aqui descrito foi desenvolvido na linguagem C, usando bibliotecas que possibilitam o uso de sockets para a comunicação entre o HTTP, da camada de aplicação, e o TCP da camada de transporte. Para seu funncionamento, é necessário configurar o navegador web para que esse utilize o proxy. Como o navegador vê o proxy como um servidor, o proxy necessita de um endereço IP e de uma porta TCP, sendo o endereço o localhost e a porta escolhida a 8001. O proxy pode também ser utilizado para filtrar conteúdo, impedindo que determinados domínios sejam acessados pelo navegador.

B. TCP

TCP é um protocolo da camada de transporte usado pelo HTTP em suas requisições e envios. Trata-se de um protocolo com conexão, onde o cliente e o servidor reconhecem que estão na conexão, com garantia de entrega dos pacotes integral e na ordem correta. Não há garantia de tempo mínimo de entrega, porém. O TCP usa o protocole de rede IP e uma porta para endereçar cada cliente e servidor. As portas TCP podem ser reservadas para funções especiais, como por exemplo a porta 80, que é a porta para requisições webs. Um servidor identifica a conexão pelo IP e pela porta, podendo mandar mais de um pocote para o mesmo cliente e identificá-las com portas diferentes, resultando em processos paralelos.

C. HTTP

HTTP, ou Hypertext Transfer Protocol, é o protocolo usado na web para receber e enviar requisições. Já estando no último nível do modelo OSI, o HTTP já recebe um objeto pronto e inteiro do TCP, e esse objeto pode ser guardado e exibido ao usuário, como, por exemplo, um arquivo HTML, uma imagem ou um áudio. O HTTP é um protocolo sem estado, isto é, informações sobre conexões passadas não são guardadas, e caso se queira guardar, é preciso fazer um cookie nas máquinas do servidor e cliente com as informações das requisições passadas. O HTTP usa sockets como interface entre si mesmo e o TCP. O HTTP pode pedir objetos de acordo com a última data de atualização do objeto, pode ajustar preferências de linguagem entre outras possibilidades de configurações.

III. Desenvolvimento

A. Inspeção

A função de inspeção alavanca o projeto do proxy, à um inspetor web, que permite a edição e análise das requisições e respostas entre o cliente e o servidor remoto. Quando o cliente faz uma requisição, a mesma é passada para o proxy e só é liberada depois de analisada e editada se for a escolha do cliente. Depois de analisada a requisição, o proxy a envia ao servidor externo correspondente. Quando este envia a resposta à requisição, o proxy intercepta, e disponibiliza ao cliente para que ele possa analisar e editar. Uma vez passada pela análise do cliente, a resposta é enviada de volta ao browser. Essa funcionalidade nos mostra uma noção da quantidade de código trocado em uma simples requisição, podendo-se destacar a probabilidade de injeção de código malicioso entre essas requisições aparentemente inocentes.

B. Dump

O Dump permite que um site seja baixado completamente, possibilitando ao cliente navegar por ele offline, usando os arquivos baixados para a pasta local. Nessa implementação em específico, o Dump funciona de forma manual, baixando uma página por vez e não recursivamente. Através do processo de desenvolvimento dessa funcionalidade podemos perceber como funcionaria a cache feita pelo browser, porém de uma forma mais estática, pois o proxy não verifica com o servidor se a página foi atualizada. E também permite que o site em

questão seja analisado a procura de código que tenha alguma função suspeita.

C. Spider

O Spider procura no arquivo HTML as expressões ”href” e ”src”, que indicam páginas web. Quando as acham, o spider extrai o conteúdo entre aspas que aparece logo a seguir e que contém um caminho de arquivos dentro de pastas, detro de pastas e um domínio, apesar de que se esse caminho for dentro do domínio do hospedeiro da coneção inicial, a indicação do domínio é desnecessário. Em seguida, cada pasta, separada pelo caractere ””, a barra, é armazenada em um estrutura de árvore, podendo-se travar um caminho seguindo a profundidade dos nós da árvore, onde cada pasta dentro de outra, aumenta a profundidade.

D. Interface Gráfica

No projeto aqui descrito, também houve a tentativa de desenvolver uma interface em GTK+, entretanto não se conseguiu unir a interface ao código do proxy. A interface possuia quatro botões, um para cafa funcionalidade, e duas saídas de texto, que seria para a requisição e para a resposta;

IV. Execução e Documentação

Para o desenvolvimento do projeto foi usada a ferramenta de versionamento GitHub. Para vizualisar os códigos e etapas, basta acessar pelo navegador o link: repositorio

A documentação do código foi feita usando a ferramenta Doxygen. Então para gerá-la, é preciso que a ferramenta esteja instalada no computador, e pelo terminal digite:

doxygen Doxyfile

Para executar o projeto, é necessário preparar o seu navegador para se conectar do browser. Nesse caso, na área de IP coloque localhost, e como porta coloque 8228, porém é possível passar como parametro a porta que deseja. Pelo terminal é digite o comando a seguir para compilar o projeto:

gcc -o aracne main.c snoopy.c snoopy_dump.c

Para a execução pode-se digitar:

./aracne

ou então:

./.aracne *porta_escolhida*

Então aparecerá um menu com as opções, como mostrado na figura 1 (Figura 1).

Ao escolher a primeira opção, o programa executará o modulo proxy puro, interceptando e dando seguimento às requisições e respostas (Figura 2).

A segunda opção é a funcionalidade de inspeção, ela receberá as requisições e mostrará a opção de editar a mesma (Figura 3).

Na terceira opção a funcionalidade do Spider irá ser executada para o site em questão, com algumas restrições (Figura 4).

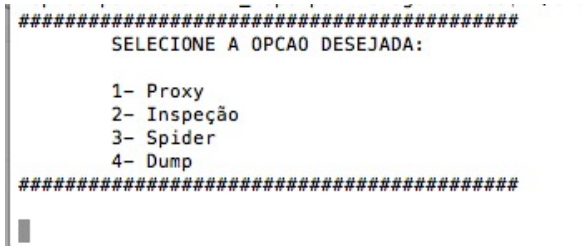

Figura 1. Opções do programa

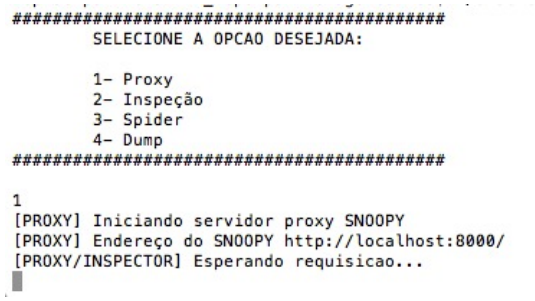

Figura 2. Execução do proxy

E na quarta e última funcionalidade, o Dump irá baixar a página cuja url foi requisitada (Figura 5):

```
##############################################
        SELECIONE A OPCAO DESEJADA:

        1- Proxy
        2- Inspeção
        3- Spider
        4- Dump
##############################################

2
[INSPECTOR] Iniciando inspector SNOOPY
[INSPECTOR] Endereço do SNOOPY http://localhost:8000/
[PROXY/INSPECTOR] Esperando requisicao...
[INSPECTOR] Requisição recebida: POST http://ocsp.pki.goog/GTSGIAG3 HTTP/1.1
Host: ocsp.pki.goog
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Length: 75
Content-Type: application/ocsp-request
Connection: keep-alive

0I0G0E0C0A0     +
[INSPECTOR] Escreva a requisição editada:
```

Figura 3. Execução do inspetor

```
##############################################
        SELECIONE A OPCAO DESEJADA:

        1- Proxy
        2- Inspeção
        3- Spider
        4- Dump
##############################################

3
[PROXY/INSPECTOR] Esperando requisicao...
```

Figura 4. Execução do Spider

```
##############################################
        SELECIONE A OPCAO DESEJADA:

        1- Proxy
        2- Inspeção
        3- Spider
        4- Dump
##############################################

4
[REQUEST] Conectando ao servidor externo...
[AGUARDE] Enviando resposta...
[AGUARDE] Resposta enviada
[AGUARDE] Conexão fechada



O arquivo index.html foi  criado
```

Figura 5. Execução do Dump