Inspetor HTTP baseado em Proxy Server

Teleinformática e Redes 2

Iago Cardoso Cortes - 13/0059943

09 de Julho de 2018

1 Apresentação teórica

1.1 Proxy Server Web

Em uma rede de computadores, um servidor proxy é qualquer sistema de computador que ofereça um serviço que atue como intermediário entre as duas partes em comunicação, o cliente e o servidor. Na presença de um servidor proxy, não há comunicação direta entre o cliente e o servidor. Em vez disso, o cliente se conecta ao servidor proxy e envia solicitações de recursos, como um documento, uma página da Web ou um arquivo que reside em um servidor remoto. O servidor proxy lida com essa solicitação, buscando os recursos necessários do servidor remoto e encaminhando os mesmos para o cliente.

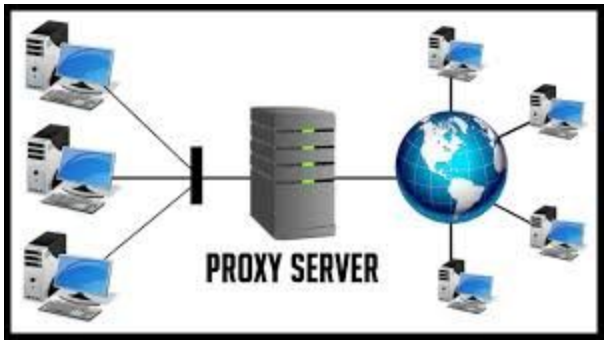


Figura 1: representação do funcionamento de um servidor proxy.

Motivos para usar servidor proxy web:

- Para compartilhar a conexão com a Internet em uma LAN.
- Para acelerar a navegação na Internet. Se usar o servidor proxy, todas as solicitações dos computadores clientes alcançarão o servidor proxy no início, se o servidor proxy armazenou em cache os recursos necessários em seu disco rígido local antes com a função de cache da Web, os clientes receberão feedback diretamente do servidor proxy. ser mais rápido do que o acesso direto.

- Para ocultar o endereço IP do computador cliente para que ele possa navegar anônimamente, isso ocorre principalmente por motivos de segurança. Um servidor proxy pode atuar como um intermediário entre o computador do usuário e a Internet para impedir ataques e acessos inesperados.
- Para implementar o controle de acesso à Internet, como autenticação para conexão com a Internet, controle de largura de banda, controle de tempo online, filtro da Web da Internet e filtro de conteúdo, etc.
- Para contornar restrições e filtros de segurança. Por exemplo, muitos escritórios de trabalho bloquearam o Facebook e o MySpace. No entanto, você pode usar o servidor proxy para contornar essas restrições e acessar facilmente sites bloqueados.
- Para verificar conteúdo de saída, por exemplo, para proteção contra vazamento de dados.
- Para contornar restrições regionais. Por exemplo, um servidor que usa a geolocalização baseada em IP para restringir seu serviço a um determinado país pode ser acessado usando um proxy localizado neste país para acessar o serviço.

1.2 TCP

O TCP (Transmission Control Protocol), é um conjunto de protocolos de comunicação usados para interconectar dispositivos de rede na Internet. Ele é o protocolo mais comumente usado na Internet.

Quando você solicita uma página da Web em seu navegador, o computador envia pacotes TCP ao endereço do servidor da Web, solicitando o envio das páginas da Web para você. O servidor da Web responde enviando um fluxo de pacotes TCP, que seu navegador da Web une para formar a página da Web. Quando você clica em um link, entra, faz um comentário ou faz qualquer outra coisa, seu

navegador envia pacotes TCP para o servidor e o servidor envia os pacotes TCP de volta.

O TCP tem tudo a ver com confiabilidade - os pacotes enviados com o TCP são rastreados para que nenhum dado seja perdido ou corrompido em trânsito. É por isso que os downloads de arquivos não são corrompidos, mesmo que haja falhas na rede. É claro que, se o destinatário estiver completamente off-line, seu computador desistirá e você verá uma mensagem de erro informando que ele não pode se comunicar com o host remoto.

O TCP consegue isso de duas maneiras. Primeiro, ordena pacotes numerando-os. Segundo, verifica o erro fazendo com que o destinatário envie uma resposta ao remetente informando que recebeu a mensagem. Se o remetente não obtiver uma resposta correta, ele poderá reenviar os pacotes para garantir que o destinatário os receba corretamente.

1.3 Protocolo HTTP

Hypertext Transfer Protocol (HTTP) é um protocolo de camada de aplicação para transmitir documentos hipermídia, como HTML. Ele foi projetado para comunicação entre navegadores da Web e servidores da Web, mas também pode ser usado para outras finalidades. O HTTP segue um modelo clássico de cliente-servidor, com um cliente abrindo uma conexão para fazer uma solicitação e aguardando até receber uma resposta. HTTP é um protocolo sem estado, o que significa que o servidor não mantém nenhum dado (estado) entre dois pedidos. Embora muitas vezes baseado em uma camada TCP / IP, ele pode ser usado em qualquer camada de transporte confiável; isto é, um protocolo que não perde mensagens silenciosamente, como o UDP.

2 Implementação e funcionamento

O trabalho foi implementado em C , e a interface gráfica escolhida foi a GTK +. Os tópicos a seguir vão discutir a arquitetura do sistema em cada uma das três funcionalidades principais propostas (proxy, spider e dump) em conjunto com a interface gráfica. A documentação do código está nos arquivos .c.
GitHub:

https://github.com/IagoCCortes/TrabTr2

2.1 Servidor proxy HTTP (http_proxy.c)

Inicialmente o programa executa com a função "execl()" um arquivo executável que lança a interface gráfica.

O servidor proxy funciona como proposto, após o browser ser configurado para utilizar uma configuração de proxy manual com endereço 127.0.0.1:8228, ou com alguma outra porta dependendo do usuário, é criado um socket utilizando a função "socket()", um endereço é especificado para este socket e é feito o binding do endereço local para o socket usando a função "bind()".

A função "listen()" é então utilizada para permitir que o proxy possa fazer conexões e quando o cliente (browser) fizer alguma requisição a mesma será aceita pela função "accept()".

O programa entra em um loop infinito para tratar qualquer requisição do browser, quando alguma ocorrer é chamada a função "getHTTP()" que recebe o request , faz o parser do mesmo, verifica se o método do request é: "GET", "POST", "PUT", "HEAD" ou "DELETE", visto que o propósito do programa é só aceitar requisições http. Alguns headers tem seus valores modificados. O header "Accept-encoding" tem seu valor modificado para "identity", para facilitar a leitura do corpo de cada response e "Connection" tem seu corpo modificado para "close".

Não foi implementada a opção de modificar os requests nem as responses. O servidor proxy que é executado como um processo diferente da interface gráfica se comunica com a mesma (enviando os requests e responses) por meio dos arquivos “request.txt” e “response.txt”. O arquivo “request.txt” é criado e é utilizado a função “kill” para forçar o sinal de sistema “SIGSTOP” no processo da proxy, para que o programa só continue após o usuário clicar no botão “Send Request” na interface gráfica que envia um sinal de sistema “SIGCONT” para o processo da proxy.

A função “serverSocket()” então é chamada é feito todo o setup para a conexão com o servidor requisitado no request usando as funções: “getaddrinfo()”, “socket()” e “connect()” e é usada a função “send()” para enviar o request para o servidor requisitado.

Naturalmente a próxima etapa é receber a resposta do servidor com a função “recv()”, é realizado um parser da mesma para facilitar a implementação da edição da response (que não deu certo) e com a response em mãos a mesma é enviada ao cliente (browser). Todo esse procedimento se repete para cada request do cliente.

2.1 Spider (spider.c)

Para executar o spider é necessário clicar no botão spider na interface gráfica, isso fará com que seja aberta uma nova janela.

Com base no header “Referer” dos requests é definida a página na qual é executado o spider. Basicamente o que ele faz é enviar um request no formato: “GET <path> HTTP/1.1\r\nHost: <host>\r\nConnection: close\r\n\r\n” o procedimento para a conexão com o servidor mencionada anteriormente é repetido e o request é enviado.

Quando a response chega, é realizado um parser da mesma na função “parseMesS()” para identificar os links. Eles foram identificados como tudo que estivesse entre: “href=“” ou “src=“” e “””.

Os mesmos são salvos em uma variável global e são mostrados na nova janela.

2.2 Dump (dump.c)

Para executar o dump é necessário clicar no botão dump na interface gráfica, isso fará com que seja aberta uma nova janela.

Com base no header “Referer” dos requests é definida a página na qual é executado o dump. Basicamente o que ele faz é enviar um request no formato: “GET <path> HTTP/1.1\r\nHost: <host>\r\nConnection: close\r\n\r\n” o procedimento para a conexão com o servidor mencionada anteriormente é repetido e o request é enviado.

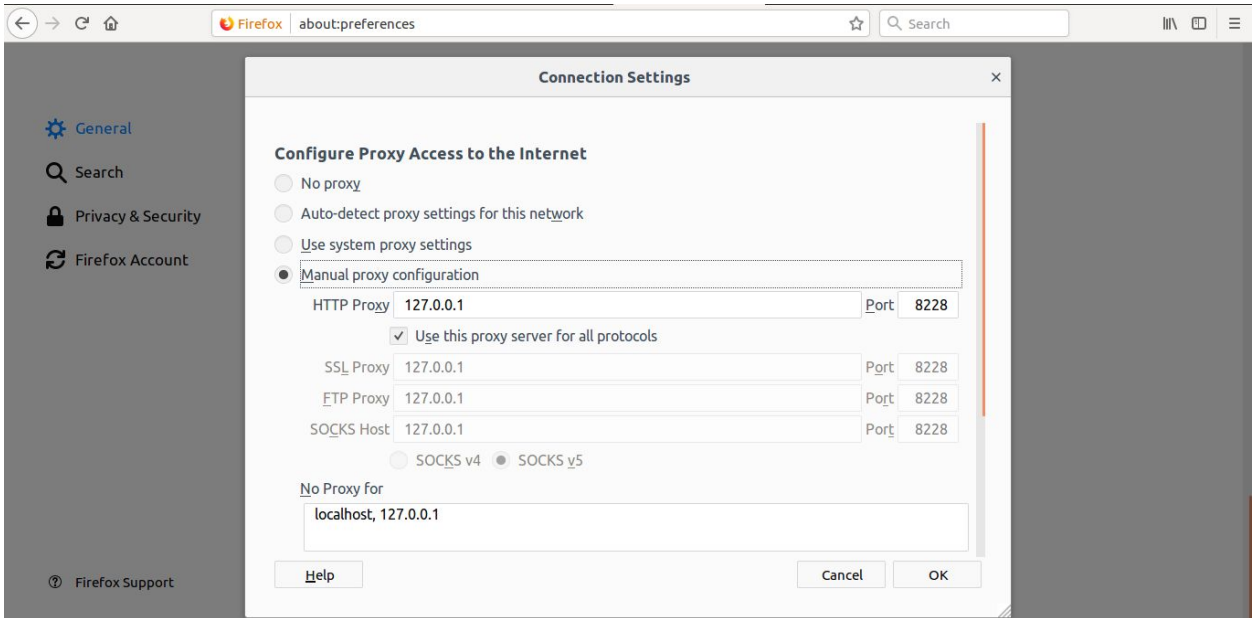
Quando a response chega, é realizado um parser do index da página na qual é realizado o dump por meio da função “parseMes()” para identificar os links e salvar o corpo da response. Eles foram identificados como tudo que estivesse entre: “href=“” ou “src=“” e “””. Os mesmos são salvos em variáveis global e são utilizados na função “createF()” que cria os diretórios se os mesmos não existirem ainda e salvam os corpos das responses em suas devidas pastas.

Para cada link identificado no index é feito um request, é extraído o path de cada arquivo baixado (corpo da response) e o mesmo é salvo na sua devida pasta com sua devida extensão (.png, .css, .js).

3 Como executar

No terminal no diretório onde se encontra o makefile dê um make. Antes de executar certifique-se que o browser está configurado corretamente. Para executar digite ./aracne [<-p port>], se for executado apenas ./aracne a porta default será 8228.

Figura 2. Configuração do firefox



4 Programa em execução

4.1 Proxy

Figura 3. Proxy pronta para enviar uma resposta à requisição à esquerda.

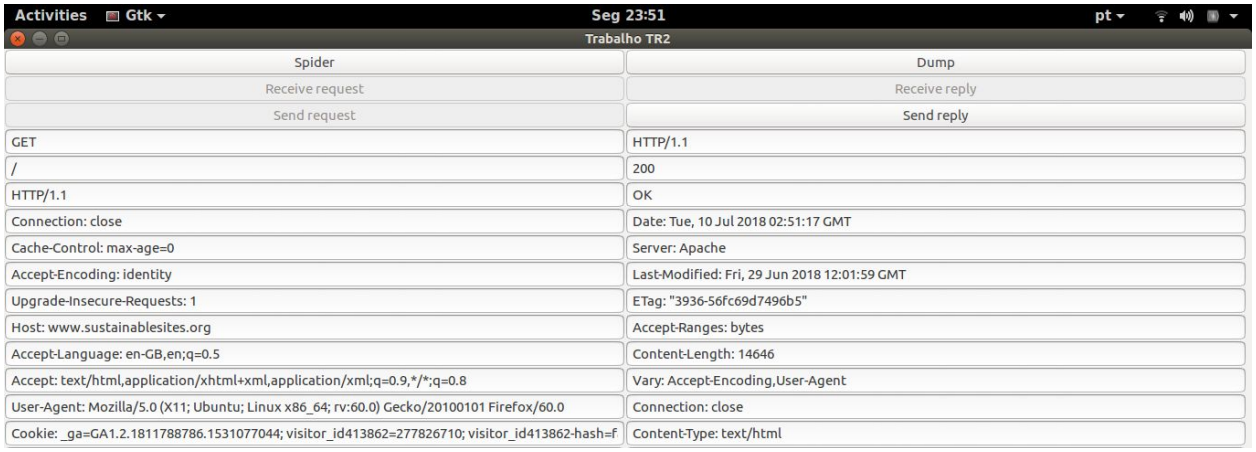


Figura 4. Proxy pronta para enviar uma requisição. À direita a resposta à última requisição.

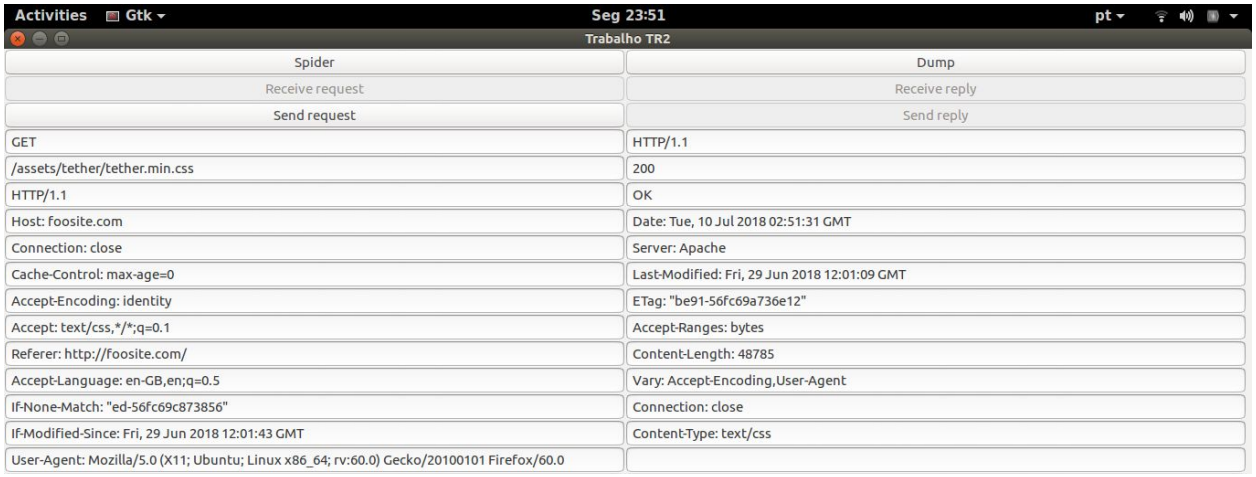


Figura 5. Proxy pronta para enviar uma resposta à requisição à esquerda.

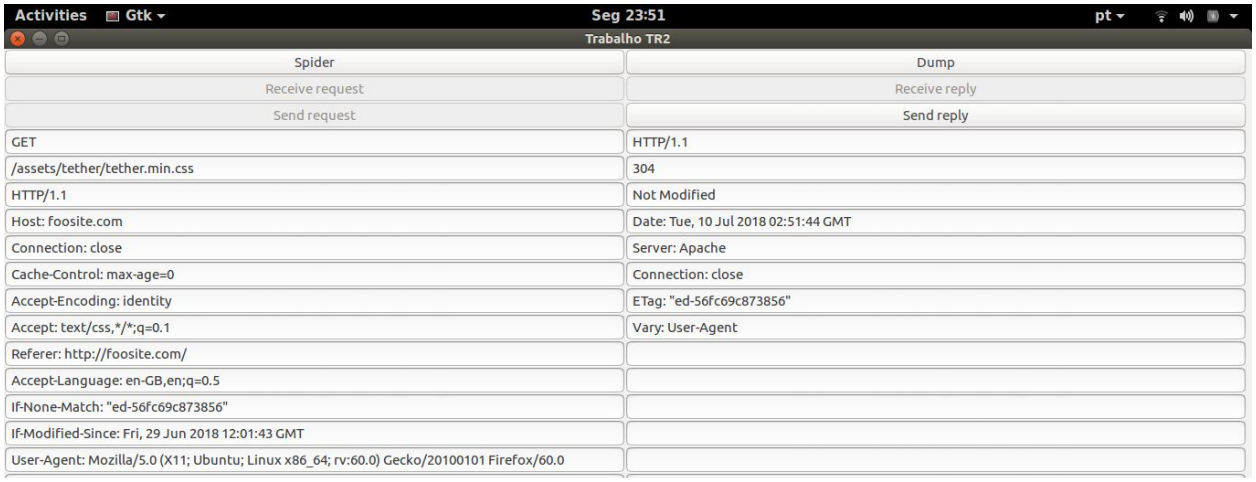
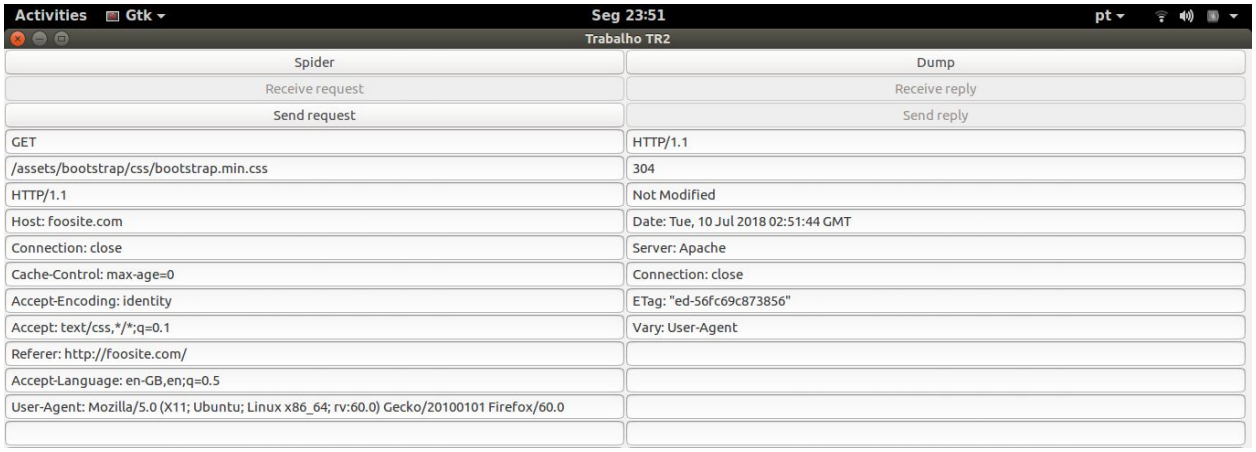
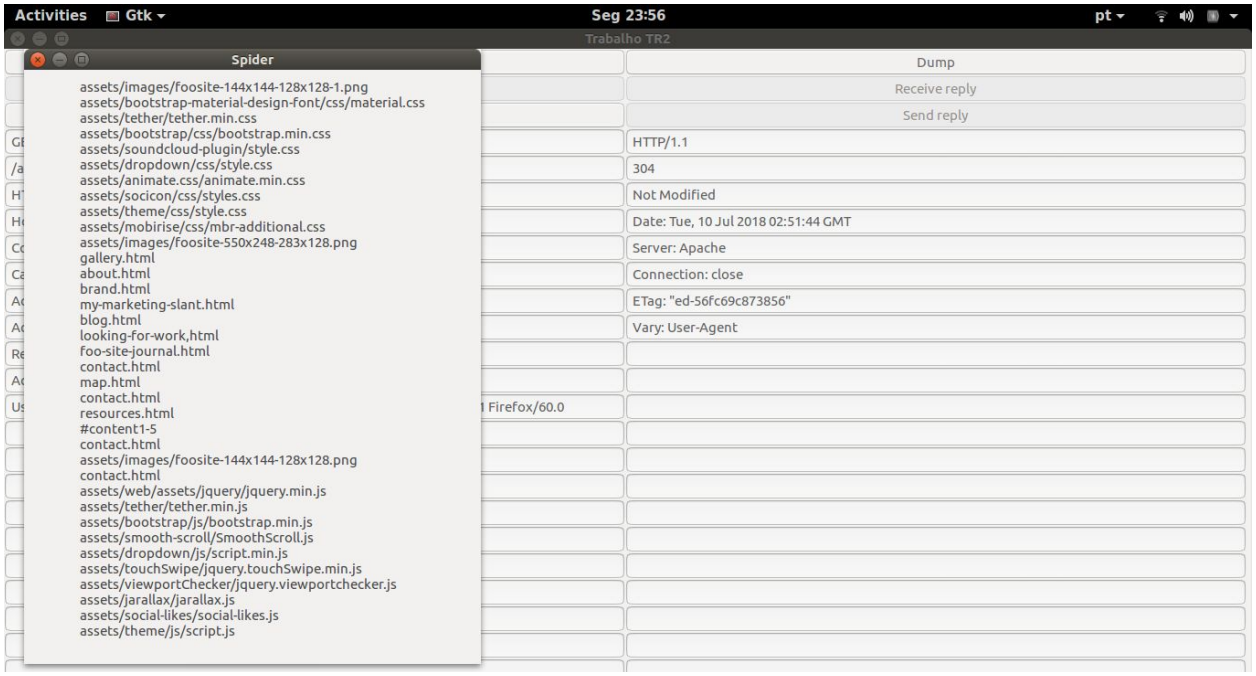


Figura 6. Proxy pronta para enviar uma requisição. À direita a resposta à última requisição.



4.2 Spider

Figura 7. Spider utilizado no site www.foosite.com



4.3 Dump

Figura 8. Dump utilizado no site www.sustainablesites.org.

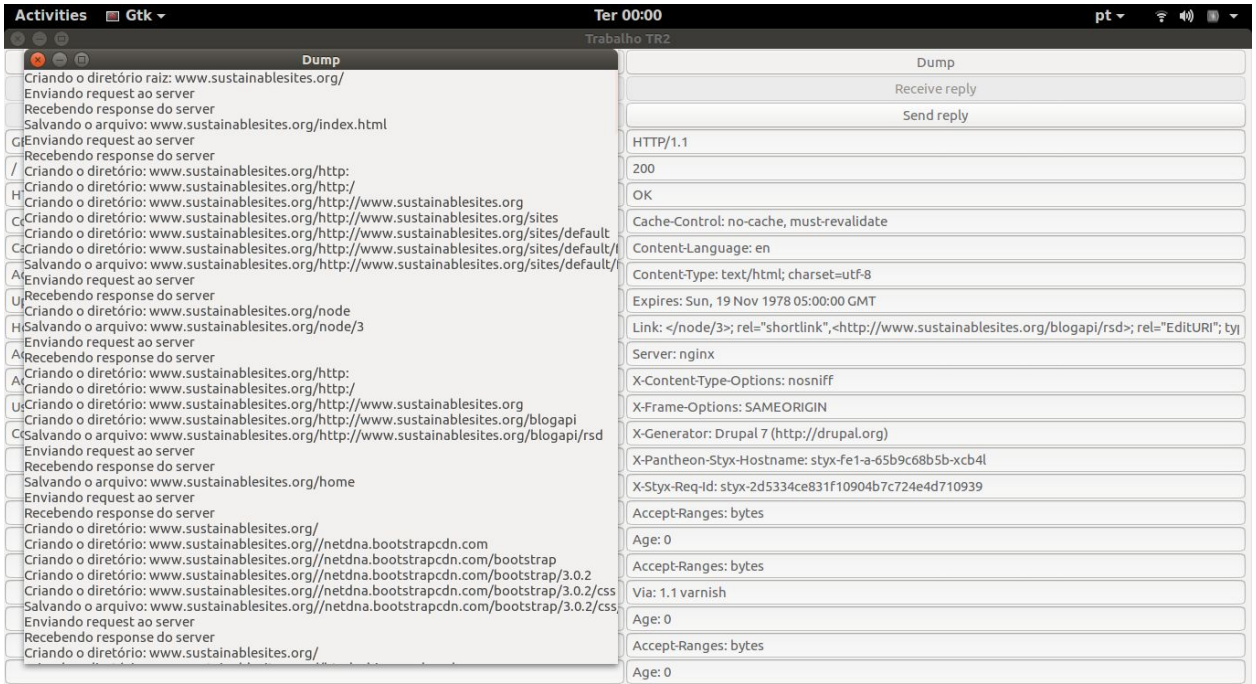


Figura 9. Arquivos e diretórios criados pelo dump utilizado no site www.sustainablesites.org.

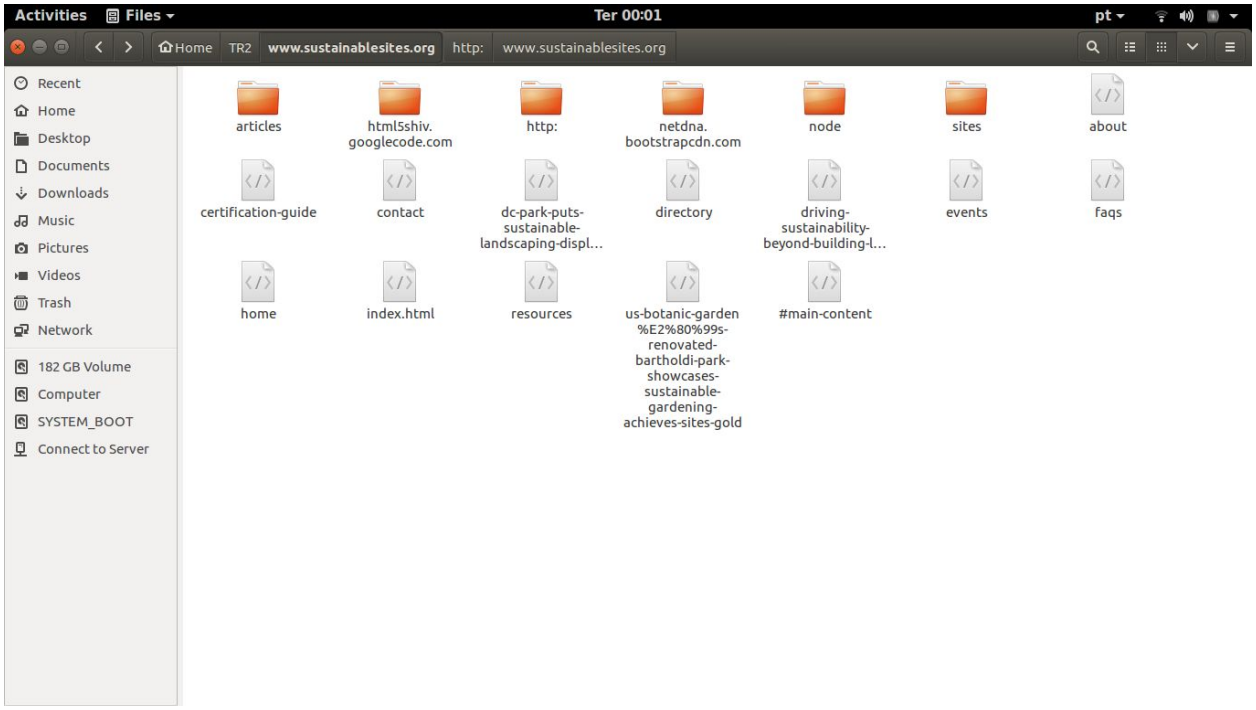


Figura 10. Resultado quando o index salvo pelo dump utilizado no site www.sustainablesites.org é aberto no browser.