Algoritmo Para Processamento De Logs CLF e Geração De Estatísticas De Acesso A Sites Por País: Visualização Dos Dados Baseada No Serviço Google Chart.

Délcio de Almeida Torres Filho, Élton Moraes, Pedro Ramos, Thiago Cabral[1] e Professor Rodrigo Cesar Vertulo[2]

Resumo:

Este artigo apresenta o desenvolvimento de um algoritmo computacional para o processamento de Logs [1] no formato CLF [2], utilizado pelo servidor de páginas de internet Apache [3]. O algoritmo permite a identificação dos países de origem de todos os usuários de sítios da internet e gera dados estatísticos sobre estes acessos, de modo que seja possível quantificar e comparar a representatividade de cada um sobre os sítios analisados. Este tipo de informação pode ser utilizada, por exemplo, para customizar o conteúdo dos sítios da internet de acordo com o idioma de origem dos países que mais o acessam, permitindo a criação de estratégias mercadológicas de acordo com cada região que apresenta maiores índices de acesso. Para o desenvolvimento foi utilizada a linguagem de Programação Python [4]. Além de o sistema apresentar estatísticas em percentual de acessos por país, também foi implementada funcionalidade de geração de mapa do mundo de forma dinâmica através da API do Google Chart [5] para apresentar, visualmente, os dados gerados pelo algoritmo. O sistema foi testado utilizando-se Logs de servidores reais da internet, bem como arquivos com endereços ip gerados de forma aleatória para testes adicionais.

Palavras-Chave: Processamento de Logs CLF; Google Chart; Python; Apache.

Abstract:

This paper presents the development of a computational algorithm for the processing of Apache Web server log files based on the CLF format. This algorithm allows the identification of the origin country of all internet web users and generates statistical data about these accesses, so that it is possible to measure and compare the weight of each one over the analyzed sites. This information may be utilized to customize web site content according to the source language of each of the accessing countries, allowing the creation of market strategies according to the regions that represent the highest levels of access. Python was the chosen programming language for the development. Not

[1] Graduandos em Tecnologia em Segurança da Informação pela Fatec São Caetano do Sul.
[2] Co-Autor / Orientador: Bacharel em Ciência da Computação pela USCS (Universidade de São Caetano do Sul), Especialista em Engenharia de Software pela Unicamp e Mestrando em Informática em Saúde pela Unifesp.

only the system presents the access rates by country in percent, but it was also implemented a dynamical world map generation to visually present the processed data by the algorithm utilizing the Google Chart API. The system was tested utilizing real internet web servers log records as well as randomly generated IP address logs for additional testing.

Keywords: CLF Log Processing; Google Chart; Python; Apache.

1. Introdução

Este artigo é um relato de experiência de um projeto desenvolvido durante a disciplina de Programação de Scripts e Verificação de Logs do curso de Tecnologia em Segurança da Informação da Fatec São Caetano do Sul no primeiro semestre de 2009.

O processamento de Logs é uma tarefa que os especialistas em redes e segurança da informação devem dominar para o bom desempenho de suas funções. A maioria dos sistemas que fornecem algum tipo de serviço remoto via rede executam o registro de cada acesso, a forma como foi feito, o que foi feito, quais operações foram realizadas, seu horário e outras informações que podem ser utilizadas para a determinação de problemas e também como evidência de que os acessos foram realizados.[3]

A correta interpretação destes registros permite ao profissional da área, investigar, analisar, determinar e propor soluções para os mais diversos problemas que podem aparecer nos sistemas, desde incidentes relacionadas à segurança, como tentativas repetidas de acesso a informações proibidas, acesso indevido de usuários, acessos em horários proibidos ou o acesso a partir de países que não deveriam acessar, até incidentes de problemas, tais como falhas, mensagens de controle ou simples registro de cada passo e atividade que os usuários realizam no sistema para auxiliar no caso de problemas.

A importância no processamento de Logs é tanta, que há um projeto de lei para colocá-lo em evidência. A questão da legalidade e obrigatoriedade do armazenamento dos Logs de acesso não afeta somente o Brasil, mas todos os países do mundo que utilizam a Internet. Há uma proposta de lei apresentada ao congresso americano para que todos que ofereçam acesso à internet mantenham os Logs de acesso e informações que possibilitem a identificação dos seus usuários.[4]

---

[3] CALDERMAN, Alexandre. LOG´s - Como interpretá-los?. Disponível em: http://www.viaseg.com.br/artigos/seguranca_logs_alexandre.htm. Acesso em: 09/09/2009.

[4] PLS - PROJETO DE LEI DO SENADO, Nº 76 , 27/03/2000, Projeto de lei que regulamenta o registro de logs, Disponível em: http://www.senado.gov.br/sf/ATIVIDADE/materia/Detalhes.asp?p_cod_mate=43555 . Acesso em: 01/06/2009.

1.1. Servidores Web

De acordo com o W3C [6], um servidor Web é o responsável por prover acesso a recursos da Web e fornecer informações ao requisitante (cliente) Web. O cliente Web é capaz de acessar recursos Web através da requisição e renderização das respostas recebidas do servidor. Uma requisição Web é uma requisição emitida por um cliente, que pode ser descrita como uma requisição Web explícita, que é manualmente iniciada pelo usuário ou requisição Web implícita, que é iniciada de forma transparente pelo cliente, sem a intervenção manual do usuário. Os recursos Web podem ser páginas em formato HTML [7], imagens, sons ou qualquer outro tipo de recurso que se queira disponibilizar para acesso através da internet por meio de um servidor Web.

1.2. Servidor Web Apache

O Apache é um servidor de páginas Web, criado de forma colaborativa cujo objetivo de desenvolvimento foi a criação de um servidor de páginas Web disponibilizado gratuitamente e desenvolvido com seu código fonte aberto, que fosse robusto, de nível comercial e com muitos recursos. [5]

O Apache gera Logs de acesso no formato CLF ou Common Log Format (Formato Padrão de Registro). Esse formato, escolhido pela maioria dos principais servidores Web, tem a seguinte estrutura:

remotehost rfc931 authuser [date] "request" status bytes

remotehost: Nome do cliente remoto (ou endereço IP se o nome do cliente não estiver disponível no DNS ou se a opção DNSLookup estiver desabilitada.

rfc931: Nome de usuário remoto do log.

Authuser: O nome de usuário, com o qual o usuário se identificou e autenticou.

[date]: Data e horário da requisição.

---

[5] APACHE, HTTP SERVER PROJECT, Disponível em: http://httpd.apache.org/ABOUT_APACHE.html[5]. Acesso em: 03/06/2009.

"request": A linha de requisição, exatamente como veio do cliente.

Status: O código de status HTTP que foi retornado ao cliente.

Bytes: O comprimento do conteúdo do documento transferido ao cliente.[6]

O Apache utiliza os Logs no formato CLF para fornecer ao administrador do sistema informações sobre sua performance, registro de acessos e informações sobre problemas.

1.3. Google Chart

O Google Chart é uma ferramenta extremamente simples que permite que você crie facilmente um gráfico em tempo real e o apresente em uma página Web. Você inclui os dados e os parâmetros para formatação em uma requisição http, e o Google retorna uma imagem em formato PNG contendo o gráfico. Muitos tipos de gráficos são suportados, e ao fazer a requisição dentro de uma tag de imagem você pode simplesmente incluir o gráfico dentro de uma página HTML.

Este serviço foi originalmente construído como uma ferramenta interna para a rápida inclusão de gráficos em aplicativos do Google, como o Google Finance, mas, como perceberam que seria uma ferramenta útil para desenvolvedores Web, foi disponibilizada gratuitamente.[7]

1.4. Materiais e Métodos

O sistema foi desenvolvido utilizando um computador com 4 GBytes de memória RAM , Processador Intel Dual Core com 2 GHz de velocidade e disco rígido de 300 GBytes.

Foram utilizados os seguintes softwares: Sistema Operacional Windows Vista com Service Pack 1 [8], Sistema Operacional Windows XP com Service Pack 3 [9],

---

[6] FORMATO DO LOG DO APACHE, Disponível em: http://httpd.apache.org/docs/1.3/logs.html. Acesso em: 01/06/2009.

[7] GUIA DO DESENVOLVEDOR, para a API do Google Chart (geração dinâmica do mapa do mundo colorido), Disponível em: http://code.google.com/intl/pt-BR/apis/chart/. Acesso em: 01/06/2009.

Ubuntu Linux 8.04 [10], Python v.2.5 [11] e a IDE de desenvolvimento WingIDE 101 v.3.1 [12].

Foi desenvolvido um programa de computador com interface em modo texto para o usuário efetuar o processamento do Log desejado utilizando o algoritmo que será apresentado adiante.

O sistema inicialmente fornece instruções sobre como escolher o Log que será processado, quer seja utilizando o sistema operacional Linux ou o Windows.

Em seguida o sistema iniciará o processamento dos dados. O Log é lido inteiramente para a memória, a partir do arquivo escolhido pelo usuário. Em seguida a base de dados que contém a referência sobre a qual país pertence cada endereço ip é carregada também na memória e seu arquivo de índice também é carregado em memória.

Então cada linha do Log é tratada para que seja extraída somente a informação do endereço IP. Este endereço IP é convertido para um número inteiro, que será utilizado para a busca na base de dados, utilizando o código otimizado, em linguagem Python, mostrado abaixo:

Tabela 01. Trecho de código de conversão de IP para número inteiro.

```
ip      = ip.split(".")
oc1     = int(int(ip[0]) << 24)
oc2     = int(int(ip[1]) << 16)
oc3     = int(int(ip[2]) << 8)
oc4     = int(int(ip[3]))
ipint   = oc1 + oc2 + oc3 + oc4
```

Utilizando-se então um algoritmo de busca binária criado pelos alunos da FATEC de São Caetano do Sul, o endereço é procurado no índice para se saber a qual faixa de endereços IP pertence. O retorno desta busca no índice é utilizado para se realizar nova busca binária na base de dados, mas desta vez restrito à faixa indicada pelo índice, o que agiliza em muito a pesquisa do país.

A sigla de dois caracteres de cada país encontrado é acrescentada em uma lista e seu nome descritivo também.

Esta lista é então percorrida e é criada uma tabela hash ou dicionário, onde a chave é a sigla do país encontrado e seu valor é incrementado cada vez que um endereço pertencente àquele país é encontrado.

O total de acessos é contabilizado através da soma de todos os acessos encontrados.

É criada outra tabela hash ou dicionário que utiliza o valor do total de acessos e o acesso por país para armazenar em cada chave (que é a própria sigla do país), o total em percentagem de acessos daquele país, encontrado no Log processado.

São criadas mais duas strings que serão utilizadas posteriormente para o download dos gráficos do Google Chart API.

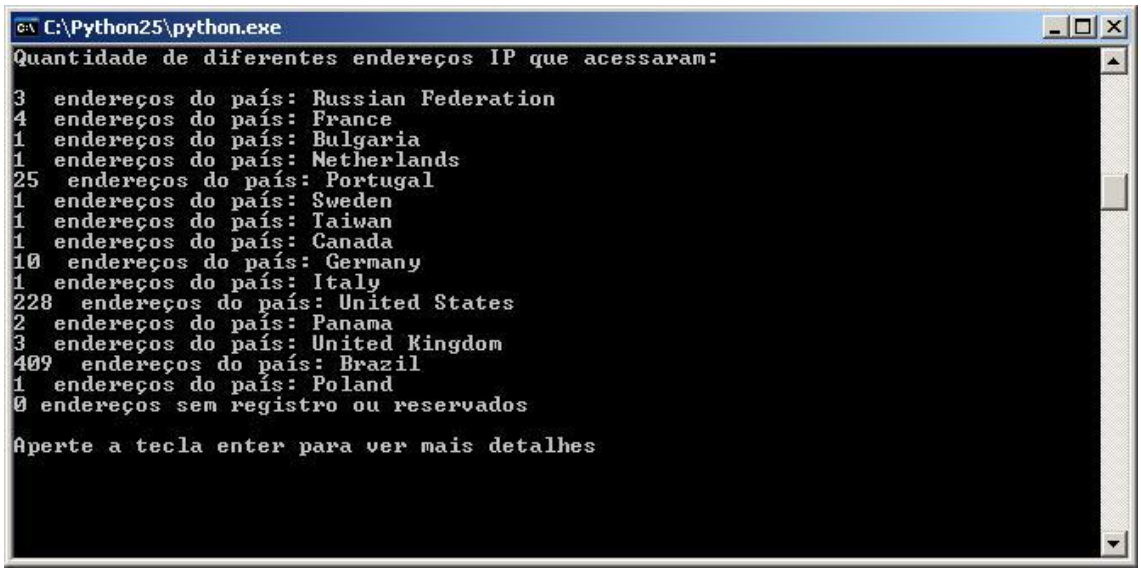
Uma string é criada percorrendo a tabela hash de acessos e armazenando a sigla de cada país encontrado, e a outra obtendo o valor do acesso percentual, formatando-o para um número float com no máximo um dígito após a vírgula e então armazenando estes valores sequencialmente separados por vírgula.

2. Resultados

Os resultados do processamento do Log são apresentados ao usuário mostrando a quantidade de acessos por país, conforme a figura 02.

O arquivo utilizado nos testes pode ser encontrado em https://sourceforge.net/projects/clfloganalysis/

Figura 02. Tela que informa quantidade de endereços ip por país.

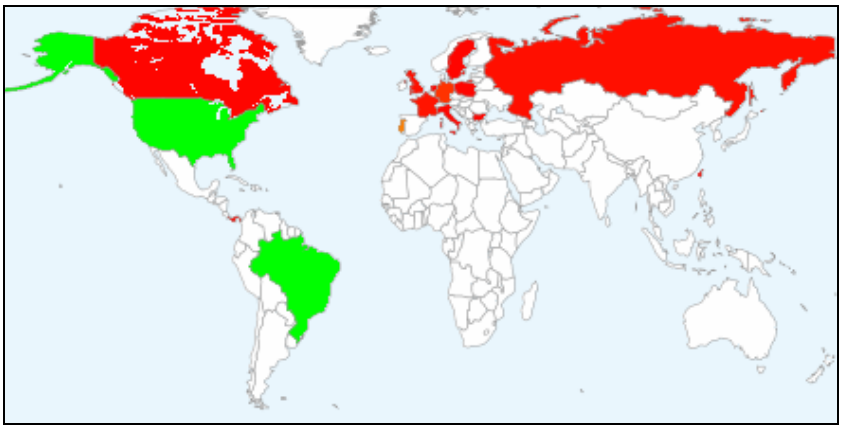


E em seguida os resultados são apresentados de forma percentual, conforme figura 03.

Figura 03. Tela que informa percentual de acesso por país.

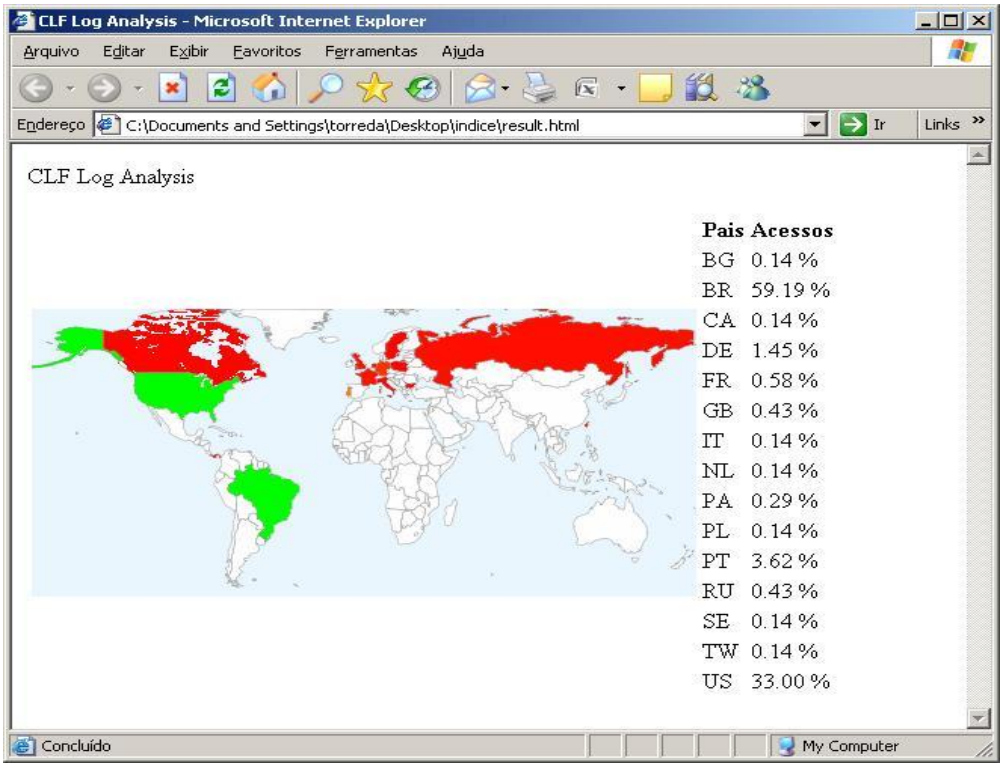
Em seguida o sistema, caso o usuário pressione a tecla enter, baixará da internet uma imagem do mapa mundial através da API Google Chart, contendo os países que acessaram o sistema, destacados por cores, conforme figura 04.

Figura 04. Mapa com os países que acessaram.



Ao final é gerado um arquivo no formato HTML que pode ser aberto em qualquer navegador de internet que mostra o mapa mundial com os países de acesso destacados e as estatísticas de acesso em formato texto, conforme a figura 5.

Figura 05. HTML contendo imagem e tabela com estatísticas de acesso.



3. Discussão

Diversas abordagens foram consideradas e testadas durante o desenvolvimento do algoritmo até que este modelo funcional atual fosse encontrado.

Inicialmente, consideramos a consulta em tempo real aos sistemas de resolução de IP por país disponíveis na internet como o WHOIS, o que mostrou-se ser inviável dado o volume de endereços IP encontrados nos Logs e também as restrições impostas à quantidade de consultas a ser realizada nos servidores WHOIS.

Um servidor WHOIS é um protocolo orientado a transações do tipo consulta/resposta, que é largamente utilizado para fornecer serviços informações aos usuários de Internet. Enquanto originalmente era utilizado como um diretório de serviços e informações sobre domínios registrados, atualmente cobre uma gama de informações maior sobre os serviços de informações disponíveis.[8]

Partimos para a busca por uma base de endereços IP versus país. Selecionamos a que continha maior quantidade de dados disponíveis e também acrescentamos endereços não alocados e privativos à tabela, disponibilizada para o uso gratuito desde que indicado que foi utilizada no sistema.

Esta base continha tanto o endereço IP em formato "string" separado por pontos, quanto o endereço IP armazenado em forma de um número inteiro. Ela foi então processada para conter somente os endereços em formato de número inteiro inicial, número inteiro final, sigla do país e sua descrição.

Optamos por utilizar os números em formato de número inteiro, por ser mais rápido ordenar uma lista de números em relação a uma lista de "strings".

Para a conversão para números inteiros, inicialmente utilizamos a forma padrão de multiplicação da linguagem de programação escolhida, mas conseguimos otimizar esta etapa através da multiplicação utilizando deslocamento de bits, conforme apresentada por BOOTH[9], o que também trouxe grande ganho no processamento. O endereço IP era obtido da linha de Log e então seus 04 octetos separados e convertidos

[8] WHOIS SERVER, Disponível em: http://tools.ietf.org/html/rfc3912. Acesso em: 20/05/2009.

[9] BOOTH, Andrew. A signed binary multiplication technique, London, 1950. Disponível em:http://bwrc.eecs.berkeley.edu/classes/icdesign/ee241_s01/PAPERS/archive/booth51.pdf/. Acesso em: 21/05/2009.

de forma individual em um número inteiro. O número final é gerado através da soma dos 04 inteiros obtidos.

Durante os testes começamos utilizando a busca linear, de modo que cada elemento da lista era percorrido e comparado com o endereço IP procurado, para descobrir a qual país pertencia o endereço, mas ela mostrou-se impraticável, conforme o tamanho dos Logs aumentava, pois para cada endereço procurado seria necessário percorrer a lista inteira, ou seja, o tempo de busca cresce de forma exponencial.

Para medir a performance, foi programada no sistema a saída para um arquivo de Log separado, contendo o horário do início do processamento e do término do processamento. Na busca linear, chegamos a obter tempos de 9 minutos para 100 mil endereços de log totalmente diferentes.

Optamos por substituir a busca linear pela busca binária na base de dados. Com isto, o tempo de processamento para 100 mil endereços caiu para 4 minutos, mas ainda não era o suficiente para nossos propósitos.

Então, optamos por criar um índice para a base de endereços IP e realizar busca binária tanto no índice quanto na base de dados, o que então trouxe a busca de 100 mil endereços para um tempo de 3 segundos.

Medidas subsequentes em arquivos maiores provaram ser esta a melhor decisão, pois em um Log com 1 milhão de endereços completamente diferentes, o tempo de processamento foi de 22 segundos e para 10 milhões de 3 minutos e 50 segundos.

Chegamos a testar a busca em base de dados “sqlite”, feita com a biblioteca “pysqlite”, mas não se mostrou tão eficiente quanto a busca indexada binária em memória.

O “sqlite” é uma biblioteca de software que implementa um motor de banco de dados transacional, auto-contido, sem necessidade de servidor e de configuração prévia. E o “pysqlite” é a ferramenta que permite a linguagem de programação Python fazer interface com o banco de dados “sqlite”[10].

[10] BIBLIOTECA PYSQLITE, Disponível em: http://trac.edgewall.org/wiki/PySqlite. Acesso em: 28/05/2009.

4. Conclusão

O sistema desenvolvido foi testado tanto em ambiente Windows quanto Linux com sucesso. A geração de gráficos também foi possível em ambos ambientes.

Vale ressaltar que para o sistema funcionar é preciso acesso à internet para a geração dos gráficos, pois é utilizado o serviço Google Chart. Para que o sistema funcionasse também em ambiente corporativo, incluímos a opção de fazer o acesso via proxy de internet dentro do código. Isto se faz necessário, pois normalmente em ambiente corporativo o acesso à internet é controlado através de software proxy, que, além de realizar o cache de informações de internet, também controla o acesso ao conteúdo para o aumento da segurança interna.

Os resultados após as diversas otimizações foram satisfatórios, contudo, o sistema poderá ser melhorado com o desenvolvimento de uma interface gráfica com o usuário e melhorando ainda mais a performance do mesmo. O sistema também poderá ser melhorado permitindo a utilização de outros padrões de Logs diferentes do CLF.

O software encontra-se disponível para download na página do projeto: https://sourceforge.net/projects/clfloganalysis

5. Glossário

[1] log – registro

[2] CLF – Common Log Format – Formato Padrão para Logs

[3] Apache – Servidor de Páginas de Internet

[4] Python – Linguagem de Programação

[5] API – Application Programming Interface - Interface de Programação de Aplicativo

[6] W3C – World Wide Web Consortium – Consórcio World Wide Web

[7] HTML – HiperText Markup Language – Linguagem para Marcação de Hipertexto

[8] – http://www.microsoft.com/windows/windows-vista/

[9] – http://www.microsoft.com/windows/windows-XP/

[10] – http://www.ubuntu.com/

[11] – http://www.python.org/

[12] – http://www.wingware.com/

6. Referências:

BOOTH, Andrew. A signed binary multiplication technique, London, 1950. Disponível em:http://bwrc.eecs.berkeley.edu/classes/icdesign/ee241_s01/PAPERS/archive/booth51.pdf/. Acesso em: 21/05/2009.

LAVOIE , Brian.Web Characterization terminology & definitions sheet. Definição de WebServer, Disponível em: http://www.w3.org/1999/05/WCA-terms/#WEBSERVERS. Acesso em: 03/06/2009.

BASE DE ENDEREÇOS DE IP, Disponível em: http://geolite.maxmind.com/download/geoip/database/GeoIPCountryCSV.zip/. Acesso em: 20/05/2009.

LICENÇA PARA REDISTRIBUIÇÃO, Disponível em: http://www.maxmind.com/app/geolitecountry. Acesso em: 20/05/2009.

BIBLIOTECA PYSQLITE, Disponível em: http://trac.edgewall.org/wiki/PySqlite. Acesso em: 28/05/2009.

USING SQLITE IN PYTON, Disponível em: http://www.devshed.com/c/a/Python/Using-SQLite-in-Python/. Acesso em: 28/05/2009.

FORMATO DO LOG DO APACHE, Disponível em: http://httpd.apache.org/docs/1.3/logs.html. Acesso em: 01/06/2009.

GUIA DO DESENVOLVEDOR, para a API do Google Chart (geração dinâmica do mapa do mundo colorido), Disponível em: http://code.google.com/intl/pt-BR/apis/chart/. Acesso em: 01/06/2009.

PLS - PROJETO DE LEI DO SENADO, Nº 76 , 27/03/2000, Projeto de lei que regulamenta o registro de logs, Disponível em: http://www.senado.gov.br/sf/ATIVIDADE/materia/Detalhes.asp?p_cod_mate=43555 . Acesso em: 01/06/2009.

APACHE, HTTP SERVER PROJECT, Disponível em: http://httpd.apache.org/ABOUT_APACHE.html[i]. Acesso em: 03/06/2009.

THE COMMON LOGFILE FORMAT, Disponível em: http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format. Acesso em: 03/06/2009.

WHOIS SERVER, Disponível em: http://tools.ietf.org/html/rfc3912. Acesso em: 20/05/2009.