UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA WEB PARA ADMINISTRAÇÃO DO SERVIDOR PROXY SQUID

VANDERSON CLAYTON SIEWERT

BLUMENAU
2007

2007/1-42

VANDERSON CLAYTON SIEWERT

FERRAMENTA WEB PARA ADMINISTRAÇÃO DO SERVIDOR PROXY SQUID

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciências da Computação — Bacharelado.

Prof. Francisco Adell Péricas, Mestre - Orientador

BLUMENAU
2007

2007/1-42

FERRAMENTA WEB PARA ADMINISTRAÇÃO DO SERVIDOR PROXY SQUID

Por

VANDERSON CLAYTON SIEWERT

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:	Prof. Francisco Adell Péricas, Mestre – Orientador, FURB
Membro:	Prof. Paulo Fernando da Silva, Mestre – FURB
Membro:	Prof. Sérgio Stringari, Mestre – FURB

Blumenau, 28 de Junho de 2007

Dedico este trabalho a Deus, a meus pais, Edson e Ivone Siewert e a minha irmã, que direta ou indiretamente me incentivaram nos momentos difíceis e sempre me deram o apoio necessário. A todos os amigos, especialmente aqueles que me ajudaram incentivando e sendo compreensivos com o tempo dedicado a realização deste.

AGRADECIMENTOS

À Deus, pelo seu amor incondicional, graça e pelo dom da vida que me foi concebido.

À minha família, que direta ou indiretamente, me incentivou para a conclusão de mais essa etapa.

Aos meus amigos, pelos empurrões, compreensão nos momentos difíceis e cobranças.

A minha tia Áurea Araldi, pelos incentivos e por acreditar nos meus sonhos e conquistas.

A Evandro José Zipf, pelas dicas e pelo auxílio prestado no desenvolvimento deste.

Aos meus avós, pela sabedoria, longas conversas e por incentivar a fazer e acontecer diferente.

Ao meu orientador, Francisco Adell Péricas, por ter acreditado na conclusão deste trabalho.

O mar é o mais sereno de todos os elementos.

Vanderson C. Siewert

RESUMO

Esse trabalho especifica e implementa uma ferramenta *web*, que permite configurar o servidor *proxy* para GNU/Linux, manipular usuários em seus grupos pré-definidos (VIP, moderado e restrito). Permite o bloqueio por palavras proibidas, por extensões de arquivos, monitoramento de *log* de acesso em tempo real e geração de relatórios de acesso a Internet.

Palavras-chave: Squid. *Proxy*. *Web*. GNU/Linux.

ABSTRACT

This work specifies and implements an web tool, that it allows to configure the server proxy for GNU/Linux, to manipulate users in its daily pay-define groups (VIP, moderate and restricted). It allows the blockade for forbidden words, extensions of archives, monitoring of log of access in real time and generation of access reports the Internet.

Key-words: Squid. *Proxy*. *Web*. GNU/Linux.

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

ACL – *Access Control List*

AD – *Active Directory*

CGI – *Common Gateway Inteface*

DAC – *Discretionary Access Control*

DHCP – *Dynamic Host Configuration Protocol*

DNS – *Domain Name System*

FTP – *File Transfer Protocol*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IMS – *If-Modified-Since*

IP – *Internet Protocol*

ISP – *Internet Service Provider*

LAN – *Local Area Network*

LDAP – *Lightweight Directory Access Protocol*

LFU – *Least Frequently Used*

LM – *Last-Modified*

LRU – *Least Recent Used*

MAC – *Mandatory Access Control*

MD5 – *Message Digest* 5

MSNT – *Microsoft New Technology*

NCSA – *National Center for Supercomputing Applications*

NT – *New Technology*

NTLM – NT *Lan Manager*

OS – *Operating System*

PAM – *Pluggable Authentication Modules*

PHP – *Hypertext PreProcessor*

RBAC – *Role-Based Access Control*

RF – Requisito Funcional

RFC – *Request For Comments*

RNF – Requisito Não Funcional

SARG – *Squid Analysis Report Generator*

SMB – *Server Message Block*

TCC – Trabalho de Conclusão de Curso

TCP/IP – *Transmition Control Protocol/Internet Protocol*

UC – *Use Case*

UML – *Unifield Modeling Language*

VIP – *Very Important Person*

SUMÁRIO

1 INTRODUÇÃO

Com o advento da Internet, o acesso à grande rede está sendo utilizado cada vez mais facilmente como ferramenta de trabalho e para fins diversos. No caso das empresas, cabe ao administrador da rede fazer o controle dos acessos à Internet, visando a segurança da rede local, fazendo o bloqueio de *sites* indesejados, de *downloads* que são ou não permitidos, entre outros.

Conforme Palma e Prates (2000, p. 9), cada vez mais os administradores têm que controlar e monitorar o acesso a recursos das redes de computadores. Com isto, surgiram ferramentas que implementam diversas funções, entre elas o filtro de pacotes, que trabalha na camada de rede[1], e os servidores *proxy*, que trabalham na camada de aplicação[2]. Estas camadas baseiam-se no modelo de referência *Transfer Control Protocol/Internet Protocol* (TCP/IP) e encontram-se descritas em Péricas (2003, p. 35).

Segundo Nemeth et al (2002, p. 44), considerando as ferramentas de administração de redes desenvolvidas para GNU/Linux em geral, especificamente em modo *console*, pode-se dizer que somente os usuários com um conhecimento mais avançado conseguem manipulá-las e usá-las apropriadamente. Conforme Pcmaster (2005), hoje já existem interfaces mais amigáveis para o usuário poder manipular as regras e estabelecer políticas de uso dos recursos da rede. Porém especificamente para os servidores *proxy*, as ferramentas são de difícil entendimento e com uma aparência nada amigável, sendo normalmente feitas em *shell script*.

A proposta deste trabalho consiste em desenvolver uma ferramenta *web* nos moldes de um *site* possibilitando a administração das políticas de acesso à Internet, grupos[3] de acessos, regras e algumas configurações do servidor *proxy* Squid, tudo isto com o aumento da segurança nas alterações do arquivo de configuração, pois qualquer administrador que não conheça o Squid será capaz de configurá-lo com simples seleções e com o preenchimento de formulários. Segundo Baros (2006), o Squid é um aplicativo que está sendo melhorado continuamente, é multi-plataforma, possui uma excelente estabilidade nas condições mais extremas e possui um imenso número de analisadores de *log*. Ele permite melhorar o

---

[1] Responsável pelo endereçamento e roteamento Internet da rede, possibilitando a conexão entre equipamentos de rede.

[2] Responsável pela comunicação entre as aplicações de rede possibilitando a transmissão de dados.

[3] Os grupos de acesso possíveis são: VIP, moderado e restrito.

desempenho de navegação na Internet com o *cache* que é armazenado localmente no servidor e implementa mecanismos de segurança nas alterações das suas configurações.

A ferramenta desenvolvida neste trabalho utilizou o servidor de páginas *web* Apache para poder interagir com o usuário em um *browser* de Internet e com o *Squid Analysis Report Generator* (SARG), desenvolvido no Brasil (ORSO, 2006), para gerar relatórios de acesso dos usuários. Ao acessar a aplicação será solicitado um nome de usuário e uma senha, que dará acesso à página liberada para fazer a administração e gerenciamento do servidor *proxy* Squid, conforme já descrito anteriormente. Foram utilizadas as seguintes tecnologias para o desenvolvimento da ferramenta: *HyperText Markup Language* (HTML), *Common Gateway Interface* (CGI) e *HyperText PreProcessor* (PHP). O emprego destas tecnologias irá melhorar a interação do usuário, através de uma ferramenta visual, com o arquivo de configuração `squid.conf` do servidor *proxy* de software livre Squid, que é onde as políticas, grupos de usuários e regras de acesso à Internet são determinadas.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver uma ferramenta visual, através de uma página *web* para fazer a gerência dirigida especificamente ao servidor *proxy* Squid.

Os objetivos específicos do trabalho são:

a) facilitar a configuração do servidor *proxy* para administradores que não conheçam o princípio de funcionamento do Squid;
b) facilitar a interpretação das políticas de utilização do servidor *proxy*;
c) definir e configurar as políticas de utilização;
d) disponibilizar a alteração de usuários nos grupos de acesso à Internet pré-determinados pelo administrador da rede;
e) implementar uma página com os parâmetros pré-configurados do Squid, para possibilitar ao usuário a configuração necessária para a sua necessidade.

1.2 ESTRUTURA DO TRABALHO

A estrutura deste trabalho está dividida em quatro capítulos, que serão explanados a seguir.

No primeiro capítulo é apresentada a introdução destacando os objetivos almejados e uma breve explicação sobre o que se pretende com este trabalho.

No segundo capítulo é apresentada a fundamentação teórica do trabalho, abordando os tópicos de gestão de redes, com alguns de seus conceitos: gerência de redes de computadores; servidor *proxy*; *cache* e seus tipos: *browse cache*, *proxy cache* e *transparent proxy cache*; os filtros do *proxy*; vantagens e desvantagens de um *proxy*; Squid; autenticação com os módulos compatíveis com o Squid; configuração do Squid; Apache; SARG; chpasswd; Webmin; controles de acesso e seus mecanismos e ACL.

No terceiro capítulo é abordado o desenvolvimento deste trabalho, com as tecnologias utilizadas e os casos de uso.

Por fim, no quarto capítulo é apresentada a conclusão do trabalho, destacando os resultados alcançados e as dificuldades encontradas.

2 GESTÃO EM REDES

Segundo Lima (1997), com o aumento da presença das redes de computadores nas instituições e como conseqüência o aumento da sua importância, faz-se necessário a gerência das redes de computadores para garantir e prevenir que alguns problemas mais graves interrompam ou prejudiquem seu desempenho e funcionalidade.

2.1 GERÊNCIA DE REDES DE COMPUTADORES

Segundo Sauvé (2002), a gerência de redes de computadores é divida em cinco partes:

a) gerência de configuração – tem por objetivo analisar, monitorar mudanças referentes à infra-estrutura física e lógica e fazer a manutenção da rede. Faz a coleta de informações de configuração de equipamentos e elementos de uma rede. Gera eventos quando recursos são agregados ou eliminados da rede, permitindo manter um inventário da rede, pois faz o registro de informações de todos os elementos que possam ser gerenciados na rede;
b) gerência de faltas – é responsável pela detecção, isolamento e resolução de falhas da rede. Através da detecção de falhas é notado algum problema nos elementos, por meio de monitoração do estado de cada um. Com o isolamento de falhas, pode-se, depois de identificada a falha, verificar a causa da falha e pode-se também fazer a antecipação das falhas, ou seja, solicitar a manutenção do elemento através de alarmes, para não prejudicar o funcionamento da rede;
c) gerência de desempenho – é responsável pela monitoração de desempenho, sua análise e pelo planejamento de capacidade. A monitoração e análise de desempenho baseiam-se basicamente em indicadores, como tempo de resposta, latência da rede, disponibilidade, taxa de erros, entre outros. O planejamento de capacidade vai basicamente demonstrar dados que sugerem a alteração no modo de operação das redes;
d) gerência de segurança – protege elementos da rede, monitorando e detectando violações da política de segurança. Preocupa-se com a proteção dos elementos da de rede, sempre com base na política de segurança pré-determinada. Faz toda a

manutenção dos *logs* de segurança para detectar violações à política de segurança;

e) gerência de contabilidade – é responsável pela contabilização e verificação de limites da utilização dos elementos de rede. Monitora quais e quantos recursos da rede estão sendo utilizados, classificando por quem e quando são utilizados. E também estabelece uma escala de tarifação.

Uma ferramenta que pode ser utilizada para gerenciar, configurar e monitorar um servidor baseado em softwares livre GNU/Linux, é o Webmin, que implementa diversas funcionalidades que permitem configurar serviços e arquivos de configuração através de uma ferramenta *web*. Atualmente não foi encontrada nenhuma ferramenta comercial que contenha estas funcionalidades para gerência de um servidor *proxy* Squid. Entretanto, esta ferramenta Webmin *freeware* e *opensource* com interface *web* que permite gerenciar vários tipos de servidores (aplicações) de rede no GNU/Linux, que entre outras funções, permite manipular algumas configurações do Squid.

Porém, segundo Pcmaster (2005), esta ferramenta não tem uma linguagem de fácil compreensão, o que dificulta a administração do servidor *proxy* por administradores inexperientes, sendo que os termos utilizados no Webmin são muito técnicos. Todos os serviços configurados pelo Webmin devem ser feitos por administradores de sistemas mais experientes, pois a ferramenta utiliza linguagem técnica que nem sempre é compreendida por um administrador menos experiente.

Mesmo a gestão de redes de computadores sendo de vital importância para as instituições, os administradores de rede ainda precisam de mecanismos com os quais possam monitorar e limitar as ações dos usuários das redes de computadores, principalmente no que se refere ao acesso a páginas de Internet.

2.2 PROXY

Em sua grande maioria, os navegadores de páginas *web*, fazem conexões diretas com a Internet. Mas há outra forma bem mais interessante de conexão: eles podem ser configurados para se conectarem através de um servidor *proxy*.

O *proxy* é um serviço que está disponível em um ambiente servidor, que recebe requisições das estações de trabalho para conexões à Internet, onde seu papel fundamental é buscar a informação primeiramente no seu *cache* local e caso não encontre o documento

requisitado, faz a busca no site solicitado pela estação de trabalho. Na segunda situação, o endereço Internet que fica registrado no servidor da página solicitada, é o do servidor *proxy*, pois o mesmo é o dispositivo que está entre a rede local e a Internet (PROXY, 2007).

Conforme Equipe Conectiva (2001), o servidor *proxy* surgiu da necessidade de ligar a rede local à grande rede de computadores, a Internet, através de um computador que provesse o compartilhamento de Internet com os demais computadores. Pode-se fazer a seguinte analogia: rede local é uma rede interna e a Internet é uma rede externa, sendo assim, o *proxy* é o dispositivo que permite as máquinas da rede interna se conectarem ao mundo externo. Como na maioria dos casos as máquinas da rede local não têm um endereço válido para a Internet, elas fazem a solicitação de um endereço externo para o servidor *proxy*, que encaminha a requisição à Internet. Caso não ache o documento solicitado em seu *cache* de Internet, o servidor está habilitado a fazer essa consulta, pois o mesmo tem um endereço válido na Internet. Sendo assim, pode-se dizer que é normal ter um servidor *proxy* diretamente ligado à Internet e com um endereço válido.

O diagrama de funcionamento do *proxy*, pode ser visto na Figura 1.

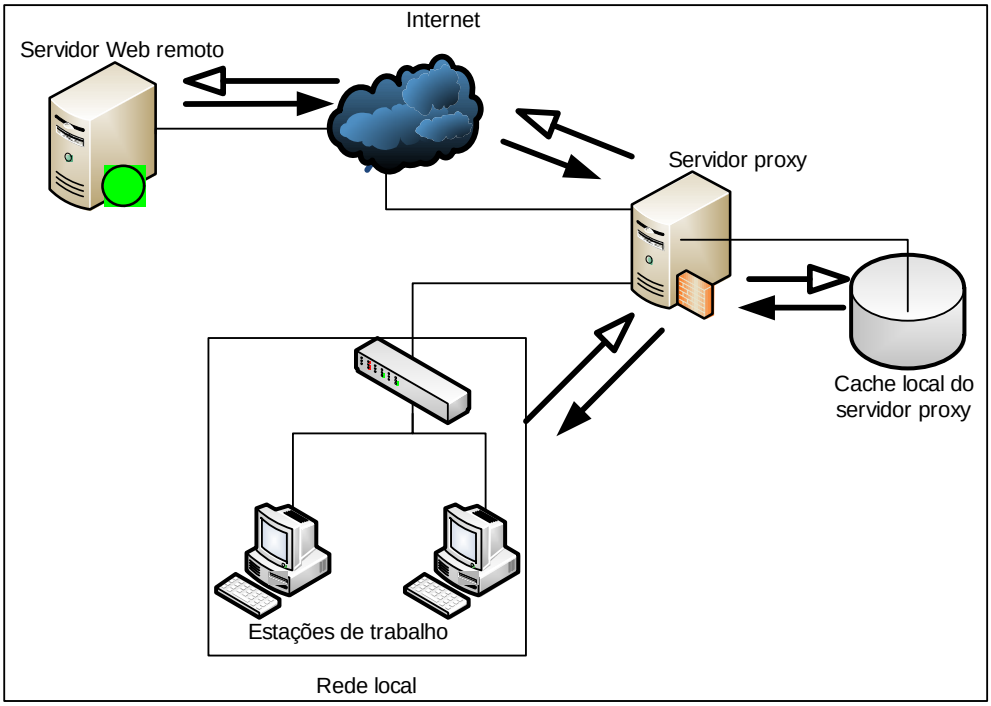

Figura 1 – Diagrama de funcionamento do *proxy*

Um dos elementos mais importante de um servidor *proxy* é o seu *cache*, além é claro dos seus filtros de bloqueio ou liberação de *sites*, as *Access Control Lists* (ACLs).

2.2.1 Cache

Conforme Watanabe (2000), o *cache* é onde os arquivos requisitados pelo servidor *proxy* são armazenados e repassados posteriormente para os clientes, que são as estações de trabalho da rede interna. Esse é um aspecto que deve ser monitorado sempre, pois pode deixar um servidor inoperante, já que são arquivos armazenados em disco e caso falte espaço em disco o servidor não vai mais funcionar. Para que isso não aconteça é necessário determinar quando os objetos[4] serão atualizados ou removidos do *cache*, sendo que alguns desses podem permanecer sem alteração alguma por tempo indeterminado e outros podem sofrer alterações frequentemente.

Conforme Proxy (2007), visando o controle do *cache*, os servidores *proxy* utilizam algoritmos de substituição que monitoram os objetos conforme seu cabeçalho, que contém a informação de período, tamanho e histórico de acessos. Dois deles são o *Least Recent Used* (LRU), que remove objetos existentes a muito tempo e o *Least Frequently Used* (LFU), que remove os objetos menos utilizados. A utilização do espaço em disco pelo *cache* do *proxy* é controlada através desses algoritmos, juntamente com regras pré-determinadas pelo administrador.

Segundo Watanabe (2000), no caso de um objeto expirado, o servidor *web* original será consultado para revalidar o objeto. Quando o objeto tem em seu cabeçalho o campo *Last-Modified* (LM), indicando qual foi sua última alteração, o *proxy* pode usá-lo para fazer a requisição *If-Modifed-Since* (IMS) ao servidor *web* remoto, fazendo a comparação da data de alteração, identificando se o objeto foi alterado ou não e poderá atualizá-lo, caso necessário, no seu *cache*. Existem três tipos de *cache*. São eles:

a) *browse cache* – conforme Watanabe (2000), a maioria dos navegadores de Internet possuem um *cache* próprio, pois é muito provável que os usuários acessem os mesmos objetos frequentemente e neste caso o *cache* não é compartilhado;
b) *proxy cache* – conforme Proxy (2007), são as implementações mais utilizadas de *proxy*, e são conhecidos também como *caching web proxy*. Este disponibiliza em *cache* páginas e arquivos de servidores remotos da Internet, permitindo que os clientes da rede local acessem de forma rápida esses arquivos, considerando que a velocidade do *link* da LAN é muito maior do que o com a Internet. Quando o

[4] Como são chamados os arquivos armazenados no *cache*.

proxy cache recebe uma solicitação de acesso a um recurso externo, como uma página da Internet, este procura primeiramente em seu *cache* local e caso não encontre o recurso solicitado, ele imediatamente faz a requisição à Internet armazenando em seu *cache* e respondendo a solicitação do cliente. Por este motivo pode-se afirmar que o *web proxy*, além de prover segurança, provê também alto desempenho para o acesso à Internet e permite criar filtros, através de regras, dizendo o que é permitido e o que é proibido. Segundo Watanabe (2000), a aplicação *proxy* age como um serviço intermediário entre as estações e os servidores remotos de Internet. Eles são utilizados por corporações que desejam reduzir a banda de comunicação que utilizam com a Internet;

c) *transparent proxy cache* – segundo Watanabe (2000), é utilizado especialmente por empresas provedoras de acesso à Internet, conhecidas como *Internet Service Provider* (ISP), porque permite o melhor aproveitamento de banda da Internet e não necessita fazer nenhuma configuração nas estações clientes. Conforme Proxy (2007), é uma forma de obrigarem os clientes a utilizarem o *proxy*, ou seja, além das características do *proxy cache*, ele implementa de forma transparente, por isso o nome, políticas de utilização e permite a coleta de dados estatísticos, entre outros. A transparência é implementada com a técnica de encaminhamento de portas, que é uma regra feita diretamente no *firewall* que faz o redirecionamento de todo o tráfego, por exemplo, HTTP, porta 80, para o *proxy*. Sendo assim não importa as configurações do usuário, pois sua utilização estará sempre condicionada a política de acesso pré-determinada. O *Request For Comments* (RFC)[5] 3040, define esse método como *proxy* interceptador.

2.2.2 Filtros do proxy

Segundo Marcelo (2005), além do *cache*, outra característica muito importante de um servidor *proxy* são os filtros que podem ser aplicados através de regras pré-determinadas pelo administrador. Dentre elas estão as restrições a sites, configuração ou não de autenticação dos usuários e controles de acesso por horário e data. Os filtros, em geral, são conhecidos em geral como ACLs.

---

[5] Documento do *Internet Engineering Task Force* (IETF), que descreve os padrões de cada protocolo da Internet.

Conforme Watanabe (2000), os administradores podem criar os filtros dos mais simples aos mais complexos, contendo regras baseadas em diversos itens, tais como:

a) endereço de rede da estação de trabalho;
b) domínio requisitado;
c) rede de origem ou destino;
d) localização do objeto requisitado;
e) período de acesso à páginas de Internet;
f) habilitar ou não a autenticação.

Todos os filtros mencionados acima podem ser utilizados sozinhos, ou então em conjunto, mas sempre lembrando que as ACLs são analisadas de forma seqüencial. Por exemplo, se existir uma ACL com duas regras, a primeira bloqueando uma determinada página da Internet e a segunda dando permissão para todas as páginas da Internet, então a primeira regra não tem função alguma, pois a última regra invalidou a primeira.

Essas ACLs são utilizadas principalmente por corporações que queiram permitir acesso a páginas que sejam de seu real interesse, conforme as regras e a política de segurança implementada na empresa.

2.2.3 Vantagens e desvantagens de um proxy

Segundo Watanabe (2000), algumas das principais vantagens de incentivar o uso de servidores *proxy*, são:

a) redução do tráfego de rede – são utilizadas menos requisições e respostas, sendo que o objeto do *cache* é recuperado, atualizado ou buscado do servidor uma única vez, o que reduz consideravelmente a utilização de banda por parte do cliente;
b) redução da carga dos servidores – são feitas menos requisições para os servidores *web* responderem. Por exemplo, diminui consideravelmente o congestionamento a esse servidores, quando há o lançamento de um novo produto;
c) redução de latência – possibilita a maior velocidade a resposta de requisições que são feitas ao objeto que está no *cache* do *proxy* e não diretamente ao servidor remoto;
d) possibilidade de acesso – considerando que a página de Internet solicitada está inacessível, se a página estiver como um objeto do *cache*, será possível responder a requisição, apenas não possibilitando a atualização da página solicitada.

Segundo Marcelo (2005), algumas das principais desvantagens na utilização de servidores *proxy*, são:

a) poucos serviços suportados – nem todos os serviços têm suporte com os *proxies* atuais, sendo assim a relação entre o cliente e o servidor *proxy* deve ser muito bem analisada;
b) atualização de configurações em clientes – carga muito grande de modificações e/ou atualizações em clientes, principalmente em redes locais com grande número de equipamentos. Em ambientes mistos o problema pode ser maior;
c) segurança em protocolos e aplicações – o *proxy* não garante a segurança de um cliente para possíveis falhas de segurança em protocolos ou aplicações, sendo assim é necessário que o *proxy* seja implementado junto a um *firewall*.

2.3 SQUID

Conforme Marcelo (2005) e Jesus (2001), o Squid é o servidor *proxy* mais utilizado atualmente na Internet, implementando todas as características já mencionadas anteriormente. Suporta os protocolos de comunicação HTTP, *File Transfer Protocol* (FTP) e *Gopher* e surgiu do projeto *Harvest* da ARPA. O nome Squid, que na tradução quer dizer lula, foi utilizado simplesmente para distinguir um projeto do outro.

Segundo Equipe Conectiva (2001, p. 134), o Squid é um servidor *proxy* para os protocolos já mencionados anteriormente. Portanto o acesso a outros serviços como, por exemplo, o correio eletrônico, deve ser configurado com a ferramenta responsável pelo filtro de pacotes, que trabalha diretamente na camada de rede.

Conforme Baros (2006), o arquivo de configuração do Squid chamado `squid.conf` está organizado em *tags* que tratam de todas as configurações, tais como porta de acesso ao servidor, programa utilizado para manipulação de senhas, tamanho e estruturação do *cache*, definição e manipulação das ACLs, que vão estipular quais são os grupos de usuários a serem utilizados, quais são os arquivos com os *sites* proibidos e/ou liberados, quais são as extensões dos *downloads* proibidas, entre outras *tags.*

O que normalmente sofre maiores alterações, e com mais freqüência, são justamente as regras definidas na *tag* ACL, onde podem ser alterados os usuários, como também as listas de *sites* proibidos e/ou liberados e as extensões dos *downloads* proibidos.

O servidor Squid pode ser obtido no seu *site* oficial *Squid web proxy cache* (CHADD, et al, 2006).

2.3.1 Autenticação

Conforme Marcelo (2005), a autenticação do Squid só pode ser habilitada se o mesmo for configurado em modo *proxy cache*. Caso seja configurado no modo *transparent proxy cache*, a autenticação não é permitida com seus módulos padrões. Segundo Vesperman (2001), os módulos de autenticação padrões do Squid são:

a) *Lightweigght Diretctory Access Protocol* (LDAP) – módulo que permite a autenticação baseada no banco de dados LDAP;
b) *Microsoft New Technology* (MSNT) – módulo que permite a autenticação baseada em um controlador de domínio Windows NT;
c) *National Center for Supercomputing Applications* (NCSA) – módulo que permite a autenticação baseado no tipo de arquivo *password* de muitos servidores *web* NCSA e segundo Marcelo (2005, p. 23) esse é o mais utilizado;
d) *Pluggable Authentication Modules* (PAM) – é um módulo de autenticação plugável e pode ser configurado para utilizar vários sistemas de autenticação;
e) *Server Message Block* (SMB) – módulo que permite a autenticação baseado em um servidor SMB tipo Microsoft NT ou Samba;
f) *New Technology Lan Manager* (NTLM) – módulo baseado em um protocolo de desafio / resposta, muito utilizado em ambientes Microsoft;
g) getpwnam – módulo baseado nos arquivos de senhas do GNU/Linux: o *passwd* e o *shadow*.

Conforme Vesperman (2001), o Squid utiliza processos auxiliares para processar as solicitações de autenticação para evitar que o mesmo seja parado ou bloqueado por causa de conexões lentas. Esses processos auxiliares são conectados por *pipes* Unix padrão e o Squid se comunica através de entradas e saídas padrão. Se o processo responder “OK”, a autenticação foi feita; se responder “ERR”, a autenticação falhou.

Como cada solicitação deve ser autenticada, o Squid guarda o nome de usuário e a senha junto com os retornos de autenticações bem sucedidas no seu *cache* por um período pré-determinado, permitindo que envie solicitações para cada página solicitando a autenticação ao usuário uma única vez.

2.3.2 Configuração

Conforme Marcelo (2005), o Squid é totalmente configurado em um arquivo chamado `squid.conf`, ou seja, toda e qualquer alteração nas ACLs ou alguma configuração específica deve ser feita nesse arquivo.

Esse arquivo contém informações como:

a) endereço de rede do servidor *proxy* e a porta de comunicação utilizada;
b) configuração para informar o tipo de *proxy*, ou seja, *proxy cache* ou *transparent proxy*;
c) qual o tamanho utilizado pelo *cache*;
d) qual rede está liberada para acessar o servidor *proxy*;
e) tipos de ACLs;
f) entre outras;

Esse arquivo é lido de forma seqüencial quando o serviço do Squid é iniciado, portanto, as ACLs são lidas da mesma forma. Caso uma ACL se refira a um arquivo externo, esse arquivo será analisado no momento que o serviço é iniciado. Sendo assim se houver alguma alteração, independente do arquivo, o Squid deverá ser reiniciado para que as novas configurações sejam aplicadas.

Para fazer o monitoramento total do Squid são necessários alguns utilitários, dentre eles o Apache, o SARG e o chpasswd.

2.3.2.1 Apache

Conforme Apache HTTP Server (2007), Apache é o servidor de páginas *web* mais utilizado no mundo, em março de 2007 o Apache era responsável pela hospedagem de 58% de todas as páginas de Internet do mundo. É compatível com sistemas GNU/Linux, Novell Netware, Microsoft, MAC OS X, entre outros sistemas operacionais.

2.3.2.2 SARG

Em Orso (2006) também é relacionado o projeto *open source* SARG. A ferramenta faz a análise dos *logs* do Squid e do *cache* do servidor *proxy*, informando ao administrador da rede onde os usuários navegaram, quanto tempo ficaram conectados, que arquivos foram baixados, qual os horários de acesso, quem foi o usuário que se autenticou, quais os *sites* proibidos que tiveram tentativas de acesso e depois gera os relatórios, que ficam disponíveis em uma página de Internet.

2.3.2.3 Chpasswd

Em Orso (2006) é relacionado o projeto *open source chpasswd*, que faz a alteração de senhas dos usuários do Squid com uma ferramenta para a *web*, contando com uma lista de outros colaboradores espalhados pelo mundo. Esta ferramenta foi desenvolvida em *perl script* e se comunica com um programa CGI, com o intuito de distribuir uma interface *web* através de um formulário para os usuários poderem alterar as suas senhas de acesso para o servidor *proxy*.

2.4 WEBMIN

Segundo Zago (2007), essa ferramenta configura diversos serviços disponíveis no GNU/Linux, utilizando apenas um navegador *web*, mas pode não contemplar todas as possibilidades de configuração, pois normalmente permite somente as diretivas padrão dos serviços, podendo ou não atender a real necessidade do administrador. Conforme o serviço que se deseja configurar e conforme sua necessidade, o mesmo pode ser inicialmente configurado no ambiente gráfico do Webmin, e terminar suas configurações e otimizações na linha de comando, o que demanda um maior conhecimento do administrador.

Em Pcmaster (2005) é relacionada a ferramenta Webmin, que faz gerência de alguns servidores (aplicações) de rede do GNU/Linux, de forma mais intuitiva para administradores de sistemas com um conhecimento mais avançado, utilizando-se de um *browser* da Internet. Ele manipula os serviços com certa restrição, ou seja, alguma configuração mais específica ou personalizada tem que ser feita diretamente no arquivo de configuração no *console*. Conforme Pcmaster (2005), a linguagem utilizada no Webmin é muito técnica, por este motivo exige um conhecimento mais avançado das configurações dos serviços. Alguns dos serviços que podem ser manipulados pelo Webmin são: *Domain Name System* (DNS) que é o servidor de nomes; *Dynamic Host Configuration Protocol* (DHCP) que distribui endereços de rede para os computadores; Apache que é o servidor de páginas para Internet; Postfix que é o servidor de mensagens eletrônicas, Samba que é o servidor de arquivos, cadastro de usuários do Samba do *console*; entre outros. Normalmente com estes serviços configurados somente são: adicionados usuários para acesso, como é o caso do Samba e Postfix; alterado o *range* de endereços *Internet Protocol* (IP), no caso do DHCP; alterado algum registro do servidor DNS; inserido algum domínio virtual no servidor Apache; entre outros.

2.5 CONTROLES DE ACESSO

Conforme Controle de Acesso (2007), controle de acesso em segurança, especificamente em segurança física de ambientes, é a permissão do acesso a recursos, salas, prédios, entre outros, a somente pessoas autorizadas. O controle físico de ambientes é feito por pessoas, meios tecnológicos, cartão de acesso, abertura de porta por meio de tranca eletrônica e/ou liberado por senha, ou mecanismos de segurança como: catracas, fechaduras, chaves, entre outros.

Segundo Campos (2006), o controle às informações deve atender ao determinado nível conforme os requisitos de segurança, sempre contribuindo com o negócio da organização. O controle de acesso na segurança da informação é baseado basicamente em três processos: autenticação, autorização e contabilidade. Assim sendo, pode-se dizer que o controle de acesso é a habilidade de permitir ou negar um objeto, sendo esse uma entidade passiva, um arquivo, um sistema, entre outros, por um sujeito, uma entidade ativa, sendo esse um usuário ou processo. A autenticação identifica quem acessou o recurso, a autorização define o que o usuário pode fazer e a contabilidade informa o que esse usuário fez:

a) autenticação e identificação – são parte de um processo de dois passos, categorizando quem pode acessar determinado sistema. No passo de identificação o usuário vai informar quem ele é, normalmente por um nome de usuário. No passo de autenticação ele vai informar uma credencial, por exemplo, uma senha;
b) autorização – define os direitos e permissões dos usuários. Esse processo é executado após a autenticação do usuário, determinando o que o usuário pode fazer no sistema;
c) contabilidade – coleta as informações de utilização dos usuários e dos recursos disponíveis a ele. Esse tipo de informação pode ser utilizada para gerenciamento, planejamento, entre outros. Existem dois tipos de contabilidade: em tempo real e a em *batch*. Na tempo real, as informações são trafegadas no momento da utilização do recurso pelo usuário; na *batch*, as informações são gravadas e enviadas após o uso, normalmente em tempos pré-determinados. As principais informações da contabilidade são a identidade do usuário, o momento de início de utilização do recurso e o seu término.

2.5.1 Mecanismos de controle de acesso

Os mecanismos de controle de acesso mais conhecidos são os baseados em identidade ou discricionários, os baseados em regras ou obrigatórios, e os baseados em papéis.

2.5.1.1 DAC

Conforme Silva (2004), o *Discretionary Access Control* (DAC) é uma política de controle de acesso baseada na permissão determinada pelo proprietário do recurso, por exemplo, um arquivo. O proprietário define quem tem acesso, qual a permissão e qual privilégio tem referente ao recurso. O Quadro 1, demonstra como são abribuídos os privilégios e as permissões dos sujeitos aos objetos.

	Objeto1	Objeto2
Sujeito1	(*read*)	(*read,write,execute*)
Sujeito2	-	(*read,write)*
Sujeito3	(*write)*	-

Quadro 1 – Matriz de acesso

Os controles discricionários podem ser utilizados empregando duas técnicas:

a) lista de controle de acesso, a ACL – é responsável por definir quais são os direitos e as permissões dos usuários sobre determinado objeto ou recurso. As ACLs possibilitam um método bastante flexível para a implementação de controles discricionários;
b) controles de acesso baseados em papéis – determina as permissões e privilégios com base no papel de determinado usuário na organização. Esse método visa a simplificação do gerenciamento das permissões e privilégios dadas aos usuários.

As permissões de acesso e direitos sobre determinados objetos são dados para qualquer grupo ou indivíduo. Um indivíduo pode pertencer a um ou mais grupos e podem adquirir permissões cumulativas ou serem eliminadas algumas permissões, dos grupos que ele não pertence.

2.5.1.2 MAC

Segundo Silva (2004), o *Mandatory Access Control* (MAC) implementa uma política obrigatória, ou seja, as regras de controle de acesso são impostas por uma autoridade central, normalmente o administrador do sistema, que especifica regras de controle de acesso para recursos e informações, garantindo que as mesmas sejam incontornáveis. Sendo assim esse mecanismo é bem mais complexo para implementar, pois utiliza política multinível e devido a sua rigidez com as regras de controle e também com relação as limitações dos seus modelos.

As políticas multinível são baseadas na classificação que estão submetidos os sujeitos e os objetos. Uma forma de viabilizar a implementação da política multinível é a sugestão de construir reticulados[6] com rótulos de segurança, sendo que os rótulos de segurança contém níveis de sensibilidade e categoria. As categorias são os compartimentos específicos do

[6] Reticulados é o conjunto matemático de elementos parcialmente ordenados.

sistema que pertencem as informações de uma determinada organização. Os níveis de sensibilidade atribuídos às informações são derivadas diretamente da classificação utilizada. Os rótulos de segurança[7] são o produto vetorial do conjunto de níveis de sensibilidade pelo conjunto de categorias, sendo que a categoria é o conjunto de todos os subconjuntos formados a partir das categorias pré-definidas no modelo.

2.5.1.3 RBAC

Segundo Silva (2004), os modelos baseados em papéis, *Role-Based Access Control* (RBAC), intermedeiam o acesso a informação baseado nas atividades que os usuários desempenham no sistema, podendo o usuário desempenhar papéis diferentes no sistema. Um papel pode ser definido como um conjunto de atividades e responsabilidades atribuídas a um cargo ou função dentro de uma organização. Sendo assim, os usuários têm autorização para exercer papéis, e os papéis recebem as permissões. A Figura 2 demonstra o modelo básico do RBAC.

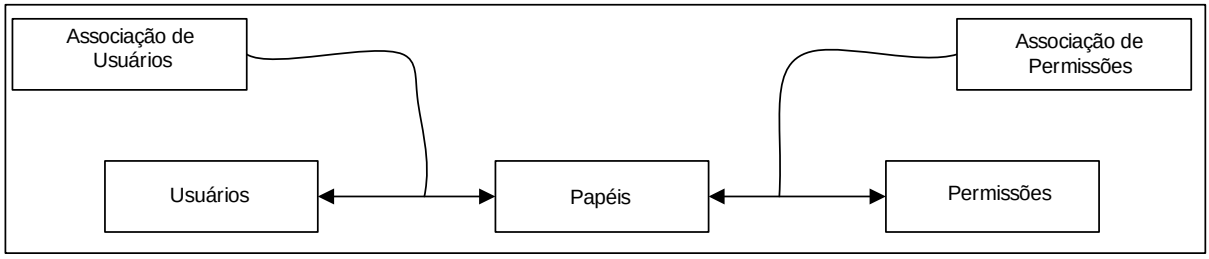

Figura 2 – Modelo básico do RBAC

O RBAC tem por objetivo facilitar a gerência de autorização, isso por que quando o usuário tem alguma mudança nas suas atribuições, sendo eliminado de um papel e atribuído a outro, a manutenção das permissões dos papéis não sofre mudanças. Normalmente o RBAC implementa o princípio de privilégio mínimo, ou seja, um usuário ativa somente o subconjunto de papéis que precisa para acessar determinado recurso ou informação e essa ativação pode ou não ter restrições.

[7] Produto vetorial do rótulo de segurança é: rótulo de segurança = nível de segurança X categoria.

2.5.2 ACL

Conforme Marcelo (2005), o *web proxy* permite ou não a autenticação, o que vai possibilitar a implementação de perfis de acesso à Internet, com o bloqueio e/ou liberação de serviços. Já o *proxy* transparente, não permite que seja configurada a autenticação, somente com algum módulo ou sistema a mais que possibilite a autenticação.

Para poder implementar esse tipo de controle por usuários é necessário a implementação de políticas de acesso, que são as populares ACLs.

Segundo Silva (2004), se for considerado uma coluna do Quadro 1, veremos que a relação de todos os sujeitos com seus respectivos direitos de acesso sobre um determinado objeto, correspondem a coluna, formando uma lista de controle de acesso, ou uma ACL, do objeto considerado. As ACLs são uma forma de representação da matriz de acesso. O Quadro 2 apresenta um conjunto de ACLs onde cada lista corresponde ao controle do objeto correspondente.

Objetos	Listas de Controle de Acesso
Objeto1	Sujeito1(*read*),Sujeito2(*write,read*),Sujeito3(*read,write,execute*)
Objeto2	Sujeito2(*read,execute*),Sujeito4(*write*)
Objeto3	Sujeito3(*read,write,execute*),Sujeito1(*execute*)

Quadro 2 – Conjunto de ACLs

A ACL de um sujeito permite uma fácil revisão dos acessos autorizados dele a um determinado objeto ou recurso. Outra operação que pode facilmente ser implementada com uma ACL é a revogação de todos os direitos de acesso de um usuário sobre um objeto, para isto basta substituir a ACL atual por uma lista vazia. Sendo assim, para determinar os acessos aos quais o sujeito está autorizado, todas as listas de controles do sistema devem ser percorridas, para fazer a revisão do acesso. A revogação de todos os acessos também requer que todas as listas de controle sejam analisadas e, eventualmente, alteradas.

3 DESENVOLVIMENTO DO TRABALHO

Neste capítulo são apresentadas técnicas e ferramentas utilizadas para a implementação da ferramenta *web*, para administração do Squid. A ferramenta desenvolvida neste trabalho é uma aplicação *web* para administração do *proxy* no GNU/Linux, baseado no servidor *proxy* Squid. A autenticação da aplicação é feita utilizando o algoritmo *Message Digest* (MD5), sendo este um recurso do servidor de páginas Apache.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

A aplicação permitirá configurar o servidor *proxy* Squid. Pode-se citar como requisitos principais da aplicação, estipulando-os em Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

a) permitir a alteração do arquivo de configuração do servidor *proxy* Squid através de formulários e múltiplas seleções utilizando *checkboxes* e *radio buttons* (RF);
b) permitir que sejam cadastrados usuários nos grupos: VIP, moderado e restrito (RF);
c) permitir que as senhas dos usuários da ferramenta sejam armazenadas no sistema pelo algoritmo MD5, que faz um *hash* da senha e é feito pelo utilitário MD5 do GNU/Linux (RF);
d) permitir que o servidor *proxy* seja reiniciado para aplicar as novas configurações (RF);
e) permitir a alteração de usuários de um grupo para outro (RF);
f) permitir a exclusão de usuários dos grupos (RF);
g) permitir o cadastro de páginas, extensões de arquivos e palavras proibidas para os grupos moderado e restrito (RF);
h) permitir o bloqueio de computadores pelos seus endereços de rede (RF);
i) permitir a liberação de páginas para o grupo restrito (RF);
j) permitir a liberação e bloqueio de portas de comunicação (RF);
k) permitir recriar o *cache* do servidor *proxy* (RF);
l) permitir a monitoração em tempo real do *log* do Squid (RF);

m) permitir a configuração dos parâmetros da aplicação estipulando onde os arquivos de configuração se encontram no servidor (RF);
n) permitir a geração e visualização de relatórios de acesso com a utilização do SARG (RNF);
o) ser disponibilizado em ambiente *web*, através do servidor Apache 2.0 (RNF);
p) ser implementado usando PHP, CGI e HTML (RNF);
q) utilizar *shell script* para integração da ferramenta com o ambiente *web* (RNF);
r) utilizar a ferramenta Macromedia Dreamweaver para edição de páginas (RNF).

3.2 ESPECIFICAÇÃO

Neste item são apresentadas as especificações da ferramenta *web*, através de *Use Case* (UC), utilizando os diagramas da *Unifield Modeling Language* (UML). Será apresentado também o diagrama de atividades da aplicação (BEZERA, 2002).

3.2.1 Caso de uso do acesso usuário ao Squid

Neste capítulo será descrito o caso de uso do servidor *proxy*, configurado pela ferramenta *web* utilizada para administrar o *proxy*, conforme descrito na Figura 3, no diagrama de caso de uso do acesso do usuário ao Squid.

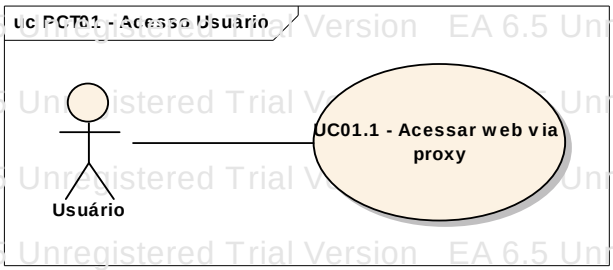

Figura 3 – Diagrama de caso de uso do acesso do usuário ao Squid

3.2.1.1 UC01.1 – Acesso web via proxy

No Quadro 3, é apresentado o caso de uso do acesso *web* via *proxy*.

Descrição	Usuário acessa a Internet da rede interna, via *proxy*, configurado pela ferramenta para administrar o *proxy*.
Ator	Usuário.
Pré-condição	Administrador deve configurar o servidor Squid pela ferramenta *web* para administrar o *proxy*.
Fluxo principal	a) verificar junto ao administrador que tipo de *proxy* foi configurado; - *proxy* autenticado; - *proxy* transparente; b) abrir o navegador de Internet; - caso o *proxy* configurado seja o autenticado, usuário deverá inserir seu nome de usuário e a senha, após seguir para passo ”c”; - caso o *proxy* configurado seja o transparente, seguir para o passo “c”; c) navegar na Internet.
Fluxo alternativo (a)	a) *proxy* autenticado: - abrir as configurações do navegador de Internet e configurar o endereço e a porta de comunicação do servidor proxy; - seguir para o passo “b” do fluxo principal. b) *proxy* transparente: - seguir para o passo “b” do fluxo principal.
Pós-condição	Usuário terá acesso a Internet.

Quadro 3 – Caso de uso acesso *web* via *proxy*

3.2.2 Casos de uso do acesso administrador

Neste capítulo serão descritos os casos de uso da ferramenta *web* para administrar o *proxy*, conforme descrito na Figura 4, no diagrama de caso de uso do acesso do administrador.

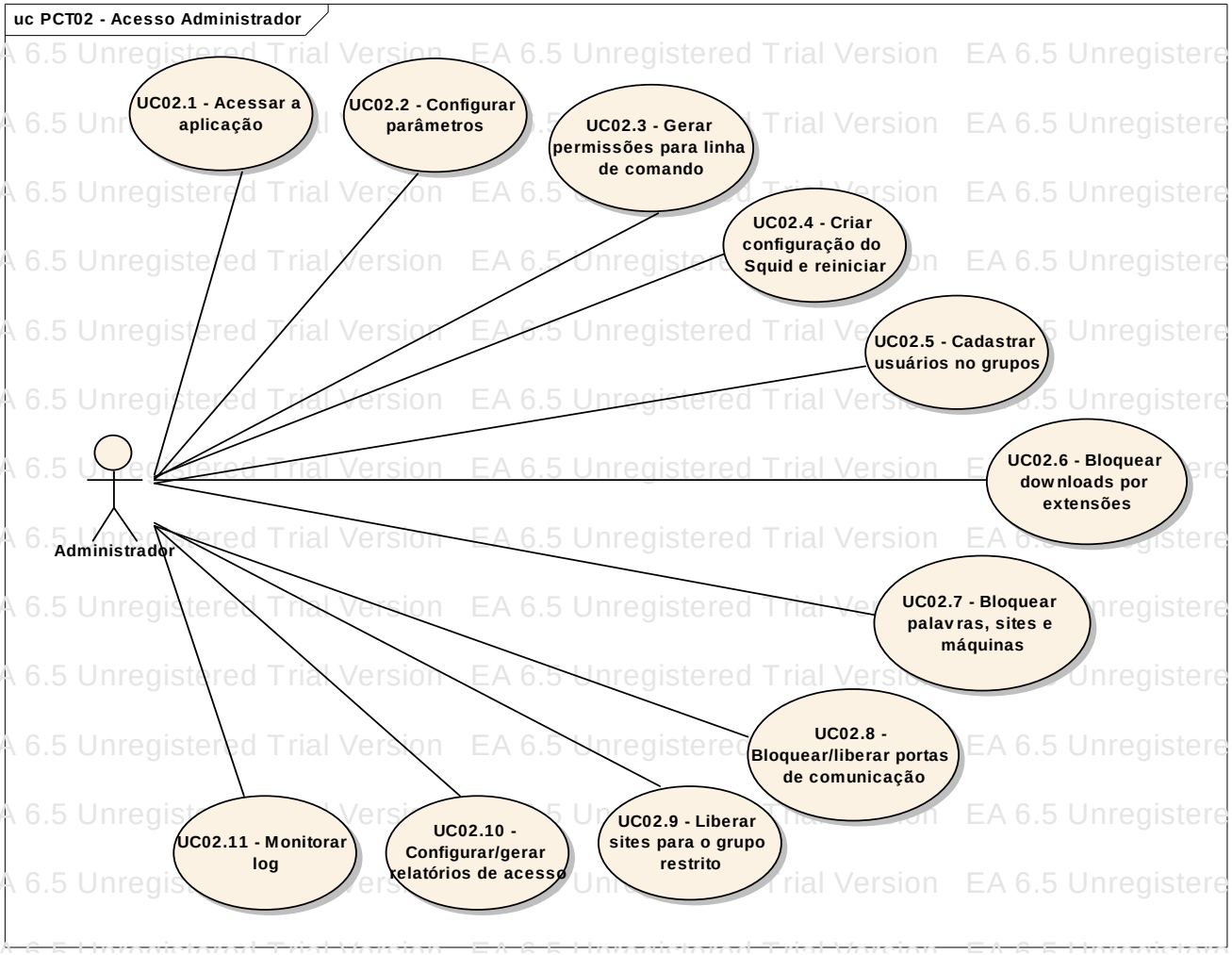

Figura 4 – Diagrama de caso de uso da ferramenta *web* para administrar o *proxy*

3.2.2.1 UC02.1 – Acessar a aplicação

No Quadro 4, é apresentado o caso de uso acessar a aplicação.

Descrição	Administrador acessa o endereço da aplicação no navegador *web*.
Pré-condição	Configuração do ambiente operacional e administrador deverá possuir nome de usuário e senha para acessar a ferramenta.
Ator	Administrador.
Fluxo principal	a) abrir o navegador Internet; b) abrir o endereço da aplicação; c) digitar o nome de usuário e senha; - usuário ou senha inválido; d) clicar no botão entrar; e) acesso a aplicação.
Fluxo exceção (c)	a) usuário ou senha inválidos: - se aparecer a mensagem "Usuário e/ou senha inválido(s)"; - fazer o passo “c” e os seguintes do fluxo principal.
Pós-condição	Administrador terá acesso a aplicação.

Quadro 4 – Caso de uso acessar a aplicação

3.2.2.2 UC02.2 – Configurar parâmetros

No Quadro 5, é apresentado o caso de uso configurar parâmetros.

Descrição	Administrador configura os parâmetros da aplicação.
Ator	Administrador.
Pré-condição	Gerar permissões para linha de comando e estar logado na ferramenta.
Fluxo principal	a) clicar no link Configuração; b) inserir o caminho absoluto do GNU/Linux, onde se encontram os aplicativos solicitados: - squid; - NCSA; - htpasswd; - SARG; - sudo; c) inserir o caminho absoluto do GNU/Linux, onde se encontram os arquivos solicitados: - *cache*; - ACL; - *log*; - arquivos de erro do Squid; d) inserir o caminho absoluto do GNU/Linux, do arquivo de senhas e para o cadastro de usuários do Squid; e) caso esteja tudo certo, clicar no botão salvar; - limpar configuração.
Fluxo exceção (e)	a) Clicar no botão Limpar; - Iniciar do passo “a”, descrito no fluxo principal.
Pós-condição	Ambiente está configurado para utilização da ferramenta.

Quadro 5 – Caso de uso configurar parâmetros

3.2.2.3 UC2.3 – Gerar permissões para linha de comando

No Quadro 6, é apresentado o caso de uso gerar permissões para linha de comando.

Descrição	Permitir gerar as permissões necessárias para a linha de comando na *console* do servidor.
Ator	Administrador.
Pré-condição	Administrador estar conectado a ferramenta.
Fluxo principal	a) clicar no *link* comandos; b) clicar no botão gerar, ao lado de permissão www; - mensagem de erro; c) permissão efetivada;
Fluxo exceção (b)	a) se for apresentada uma mensagem de erro; b) fazer o passo "a" do fluxo principal e os seguintes, até não apresentar erro.
Pós-condição	Configurado permissões para executar os comandos na *console*.

Quadro 6 – Caso de uso gerar permissões na linha de comando

3.2.2.4 UC2.4 – Criar configuração do Squid e reiniciar

No Quadro 7, é apresentado o caso de uso criar configuração do Squid e reiniciar.

Descrição	Permitir configurar o Squid, criar o *cache* e reiniciar o mesmo para que as alterações tenham efeito.
Ator	Administrador.
Pré-condição	Criar permissões para linha de comando e configurar os parâmetros, conforme ambiente operacional
Fluxo principal	a) clicar no link criar; b) no campo IP/Porta, inserir o endereço da rede interna do servidor *proxy* e a porta de comunicação utilizada; c) selecionar o tipo de *proxy*: - *proxy* autenticado; - *proxy* transparente;

	d) no campo tamanho do *cache* em *bytes*, inserir o tamanho do diretório de *cache* do servidor *proxy*; e) no campo rede liberada / máscara, inserir o endereço de rede interna, com sua respectiva máscara, que será liberada para acessar a Internet; f) no campo bloqueio de *downloads*, marcar a seleção se deseja configurar o bloqueio de *downloads* da Internet; g) no campo bloqueio de *sites*, marcar a seleção se deseja configurar o bloqueio de *sites* à Internet; h) no campo bloqueio de palavras chaves, marcar a seleção se deseja configurar o bloqueio de palavras chaves à Internet; i) no campo bloqueio de computadores, marcar a seleção se deseja configurar o bloqueio de computadores à Internet da rede interna; j) clicar no botão gerar script; k) clicar no link comandos; - clicar no botão criar cache; l) clicar no botão reconfigura o Squid.
Fluxo alternativo (c)	a) *proxy* autenticado: - será configurado o Squid com o módulo de autenticação; - será utilizado o módulo de autenticação NCSA. b) *proxy* transparente: - será configurado o Squid para ser um *proxy* transparente; - deverão ser configuradas regras para o *proxy* transparente no *firewall*.
Fluxo alternativo (k)	a) clicar no *link* comandos; b) clicar no botão gerar, da opção criar *cache*; c) caso seja a primeira configuração, o *cache* do Squid deverá ser criado.
Pós-condição	Squid configurado.

Quadro 7 – Caso de uso criar configuração do Squid e reiniciar

3.2.2.5 UC2.5 – Cadastrar usuários nos grupos

No Quadro 8, é apresentado o caso de uso cadastrar usuários nos grupos.

Descrição	Permitir o cadastro de usuários nos devidos grupos de acesso. Permitir a consulta de usuários nos grupos de acesso e a alteração de grupos de acesso.
Ator	Administrador.
Pré-condição	Squid configurado.
Fluxo principal	a) clicar no *link* usuários; b) clicar no *link*: - cadastro de usuários; consulta / alteração de usuários;
Fluxo alternativo (b)	a) cadastro de usuários: - no campo nome / senha, inserir o nome de usuário e sua senha de acesso à Internet; - selecionar o grupo de usuários a que pertence; - clicar no botão cria usuário; - será apresentada uma mensagem: “Usuário: nome do usuário, inserido no grupo de usuários selecionado com sucesso”. b) consulta / alteração de usuários: - consultar o usuário, basta acessar esse *link*; - alterar o usuário de grupo: - selecionar o usuário a ser apagado do grupo; - apagar o nome de usuário selecionado, com a tecla *delete*; - usuário será apagado do grupo; - clique no botão voltar; - clicar no botão atualizar nome do grupo escolhido; - inserir o mesmo nome de usuário que foi excluído anteriormente, no grupo escolhido; - clicar no botão atualizar nome do grupo escolhido; - clique no botão voltar; - usuário será mostrado no grupo que foi inserido.

Fluxo exceção (a)	a) Se for apresentada mensagem de erro: - "Por favor preencha o campo usuário"; - "Por favor preencha o campo senha"; - "Por favor selecione o grupo desejado"; b) fazer o passo "b" do fluxo principal., e os seguintes até não apresentar erro.
Pós-condição	Usuários inseridos em seus grupos de acesso conforme política de acesso à Internet.

Quadro 8 – Caso de uso cadastrar usuários nos grupos

3.2.2.6 UC2.6 – Bloquear downloads por extensões

No Quadro 9 é apresentado o caso de uso bloquear *downloads* por extensões.

Descrição	Permitir o bloqueio de *downloads* filtrados pelas extensões dos arquivos, tanto do grupo de usuários moderados como do grupo de usuários restritos.
Ator	Administrador.
Pré-condição	Squid configurado para bloquear *downloads.*
Fluxo principal	a) clicar no *link downloads*; b) inserir as extensões de arquivos que devem ser bloqueadas na área de texto, uma abaixo da outra; clicar no botão salvar.
Pós-condição	Squid configurado para bloquear *downloads* conforme lista de extensões definida.

Quadro 9 – Caso de uso bloquear *downloads* por extensões

3.2.2.7 UC2.7 – Bloquear palavras, sites e máquinas

No Quadro 10 é apresentado o caso de uso bloquear palavras, *sites* e máquinas.

Descrição	Permitir o bloqueio de palavras e *sites* para o grupo de usuários moderado e também permitir o bloqueio de máquinas pelo seu endereço de rede.
Ator	Administrador.
Pré-condição	Squid configurado para bloquear palavras, sites e máquinas.
Fluxo principal	a) palavras; b) *sites*; c) máquinas.
Fluxo alternativo (a)	a) clicar no *link* palavras; b) inserir as palavras que devem ser bloqueadas na área de texto, uma abaixo da outra; c) clicar no botão salvar.
Fluxo alternativo (b)	a) clicar no *link sites*; b) inserir os *sites* ou parte do endereço que devem ser bloqueados na área de texto, um abaixo da outro; c) clicar no botão salvar.
Fluxo alternativo (c)	a) clicar no *link* máquinas; b) inserir os endereços de rede que devem ser bloqueados na área de texto, um abaixo da outro; c) clicar no botão salvar.
Pós-condição	Squid configurado para bloquear palavras proibidas, *sites* proibidos e máquinas não permitidas ao acesso à Internet, conforme lista inserida.

Quadro 10 – Caso de uso bloquear palavras, *sites* e máquinas

3.2.2.8 UC2.8 – Bloquear/liberar portas de comunicação

No Quadro 11 é apresentado o caso de uso bloquear/liberar portas de comunicação.

Descrição	Fazer a liberação ou bloqueio de portas de comunicação que são utilizadas em aplicações no navegador de Internet e fazem as requisições em cima da porta de comunicação padrão da Internet.
Ator	Administrador
Pré-condição	Squid configurado para bloquear/liberar portas de comunicação utilizadas diretamente no navegador de Internet.
Fluxo principal	a) clicar no *link* portas; b) liberar portas; c) bloquear portas.
Fluxo alternativo (b)	a) inserir no campo liberar porta, o número da porta de comunicação a ser liberada pelo *proxy*; b) clicar no botão liberar; consultar as portas de comunicação liberadas na área de texto das portas liberadas.
Fluxo alternativo (c)	a) inserir no campo bloquear porta, o número da porta de comunicação a ser bloqueada pelo *proxy*; b) clicar no botão bloquear; c) consultar as portas de comunicação liberadas na área de texto das portas bloqueadas.
Pós-condição	Squid configurado para bloquear/liberar portas de comunicação utilizadas diretamente no navegador de Internet, conforme lista inserida de portas bloqueadas e liberadas.

Quadro 11 – Caso de uso bloquear/liberar portas de comunicação

3.2.2.9 UC2.9 – Liberar sites para grupo restrito

No Quadro 12 é apresentado o caso de uso liberar *sites* para grupo restrito.

Descrição	Permitir a liberação de *sites* na *web* para usuários do grupo restrito.
Ator	Administrador.
Pré-condição	Squid configurado.
Fluxo principal	a) clicar no *link* domínios; b) inserir parte ou o endereço completo dos *sites* que devem ser bloqueados na área de texto, um abaixo do outro; c) clicar no botão salvar.
Pós-condição	Lista de *sites* liberados para o grupo de usuários restritos.

Quadro 12 – Caso de uso liberar *sites* para grupo restrito

3.2.2.10 UC2.10 – Configurar/gerar relatórios de acesso

No Quadro 13 é apresentado o caso de uso configurar/gerar relatórios de acesso.

Descrição	Configuração e geração dos relatórios de acesso à Internet.
Ator	Administrador.
Pré-condição	Configurar parâmetros, gerar permissões para linha de comando e o administrador deve estar conectado a ferramenta.
Fluxo principal	a) clicar no link relatórios; b) inserir no campo caminho relatórios o caminho absoluto de onde os relatórios serão gerados; c) inserir no campo título do relatório o título do relatório para quando o mesmo for gerado; d) clicar no botão Alterar; e) clicar no botão gravar sarg.conf: - gerar relatórios; - consultar relatórios.

Fluxo alternativo (e)	a) gerar relatórios: - clicar no *link* comandos; - clicar no botão gerar, da opção gerar relatórios; b) consultar relatório: - clicar no *link* comandos; - clicar no *link* consulta relatórios; - selecionar o relatório desejado;
Pós-condição	Configurado SARG, para gerar os relatórios de acesso à Internet pelo *proxy*. Relatórios prontos para serem gerados.

Quadro 13 – Caso de uso configurar/gerar relatórios de acesso

3.2.2.11 UC2.11 – Monitorar log

No Quadro 14 é apresentado o caso de uso monitorar *log*.

Descrição	Permitir acesso ao *log* do Squid em tempo real.
Ator	Administrador
Pré-condição	Squid configurado e em funcionamento.
Fluxo principal	a) clicar no *link* monitoração; b) será mostrado o arquivo de *log* em tempo real, na área de texto.
Pós-condição	Análise do *log* do Squid.

Quadro 14 – Caso de uso monitorar *log*

3.2.3 Diagrama de atividades

O diagrama de atividades apresentado na Figura 5, demonstra a interação da aplicação com o usuário da rede interna, que deseja acessar a Internet e o diagrama apresentado na figura 6, demonstra como o administrador de rede administra e gerencia o servidor *proxy*. Os controles de acesso são definidos pelo administrador de rede conforme a política elaborada anteriormente.

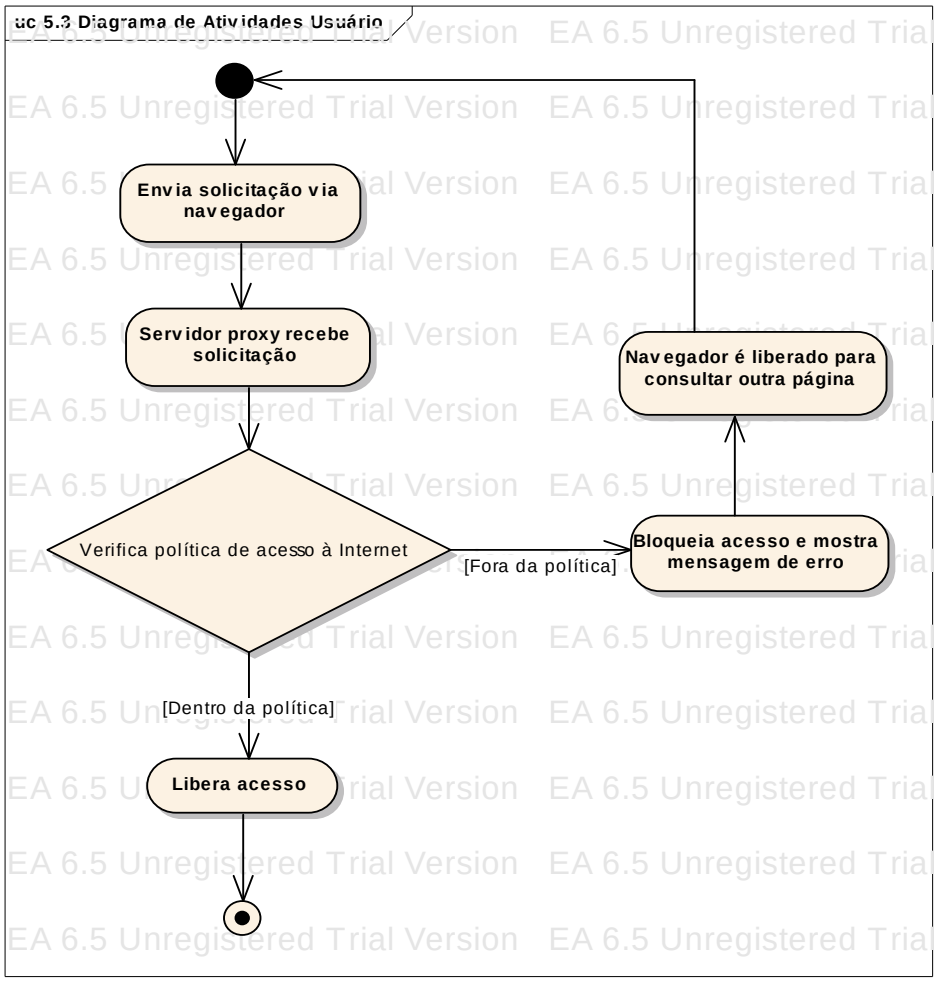

Figura 5 – Diagrama de atividades do usuário da rede interna

No diagrama de atividades do usuário da rede interna, está descrito o funcionamento e direção das requisições de acesso à Internet que o mesmo solicita. Sendo assim, cabe ao servidor *proxy*, liberar ou não o acesso conforme política de acesso definida pelo administrador de rede.

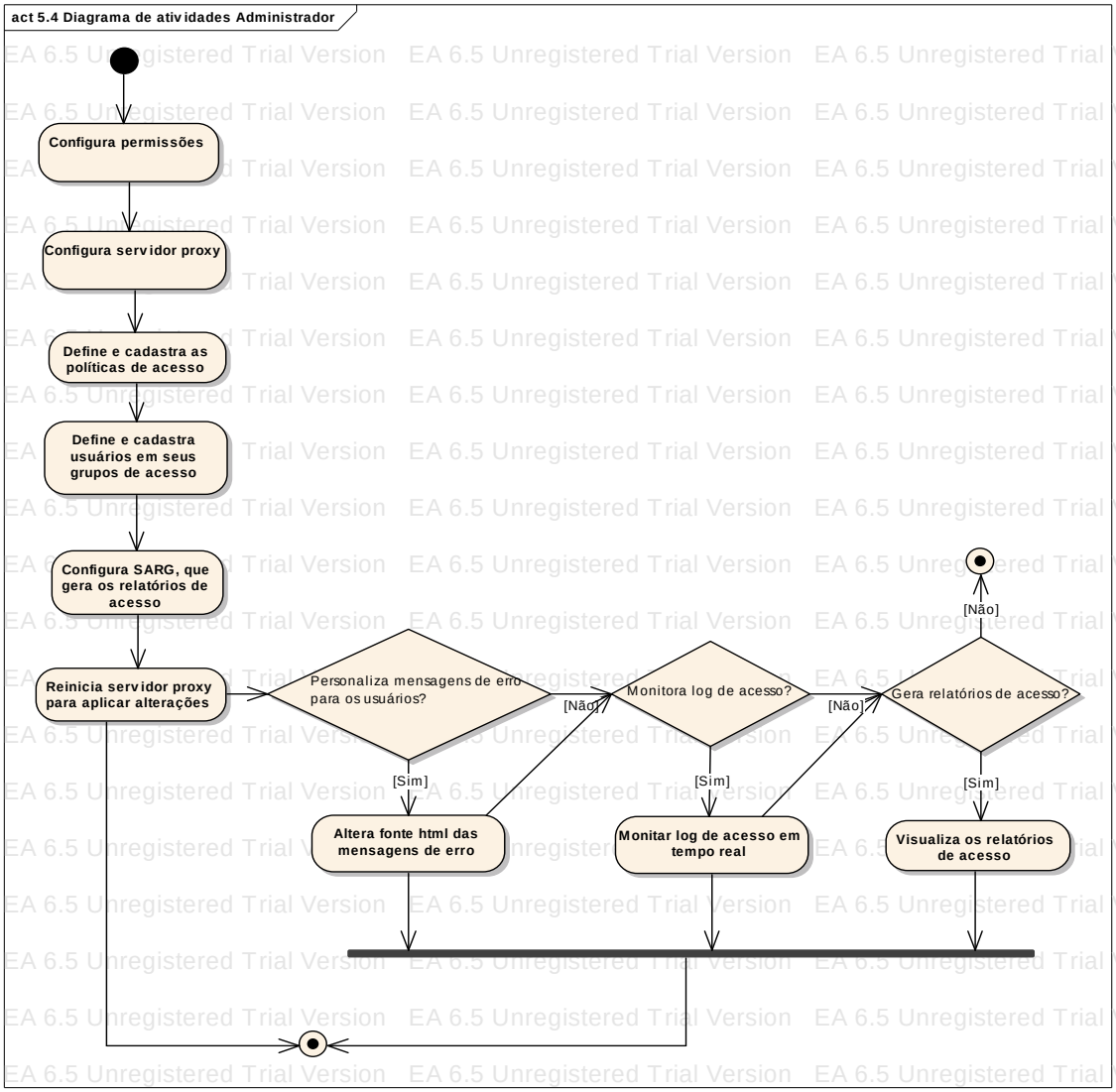

Figura 6 – Diagrama de atividades do administrador de rede

Na Figura 6, são demonstradas as atividades que o administrador de rede tem para poder implementar as políticas de controle de acesso, configurar o servidor *proxy* Squid, acompanhar os *logs* de acesso em tempo real, configurar e gerar os relatórios de acesso à Internet dos seus usuários da rede interna, alterar mensagens de erro que são apresentadas aos usuários em caso de bloqueio de acesso por tentarem acessar alguma página que vai contra a política definida e também permite reiniciar o Squid para aplicar as alterações feitas.

3.3 IMPLEMENTAÇÃO

Este item abordará as técnicas e o desenvolvimento da ferramenta *web* para

administração do Squid. Na implementação da ferramenta as linguagens de programação PHP, HTML, CGI e *shell script*. Para o desenvolvimento do *layout* da página foi utilizado o Macromedia Dreamwaver. Foi feita a integração com a ferramenta SARG, para que a mesma possa analisar os *logs* do Squid e gerar os relatórios de acesso do servidor *proxy*. Isto foi implementado com o auxílio do *sudo*, para poder executar o SARG no *console* para que os relatórios sejam gerados na forma de páginas da *web*, pois somente é executado em linha de comando.

As validações e testes serão feitos na forma de simulações, conforme os casos de uso, de usabilidade e qualidade da ferramenta *web* para administração do servidor *proxy* Squid.

3.3.1 Técnicas e ferramentas utilizadas

As ferramentas utilizadas na implementação foram as linguagens de programação PHP, HTML e java *script*. Foi utilizado o servidor de páginas Apache, para fazer a interação da ferramenta com a *console* do servidor, foi utilizado a ferramenta sudo, que permite a execução de comandos para usuários comuns do sistema operacional, com direitos de super usuário, tudo isso utilizando o ambiente GNU/Linux.

Para viabilizar a implementação da ferramenta para administração do servidor *proxy*, foi analisada a ferramenta Webmin com as suas funcionalidades e praticidade de utilização, sua linguagem, feito o estudo detalhado dos requisitos do sistema e dos casos de uso. A implementação utilizando uma ferramenta com interface *web*, foi um dos pré-requisitos, pois a mesma está presente em quase todas as ferramentas de administração atualmente. Sem contar com a facilidade de acesso e interação com o usuário, no caso o administrador de rede, que irá utilizar a ferramenta para criar as políticas de acesso à Internet, configurar o *proxy* da rede e também efetuar os cadastros necessários para a estrutura. A Figura 7 demonstra as requisições feitas tanto pelos usuários como pelo administrador de rede e também o funcionamento e interação da ferramenta com o Squid.

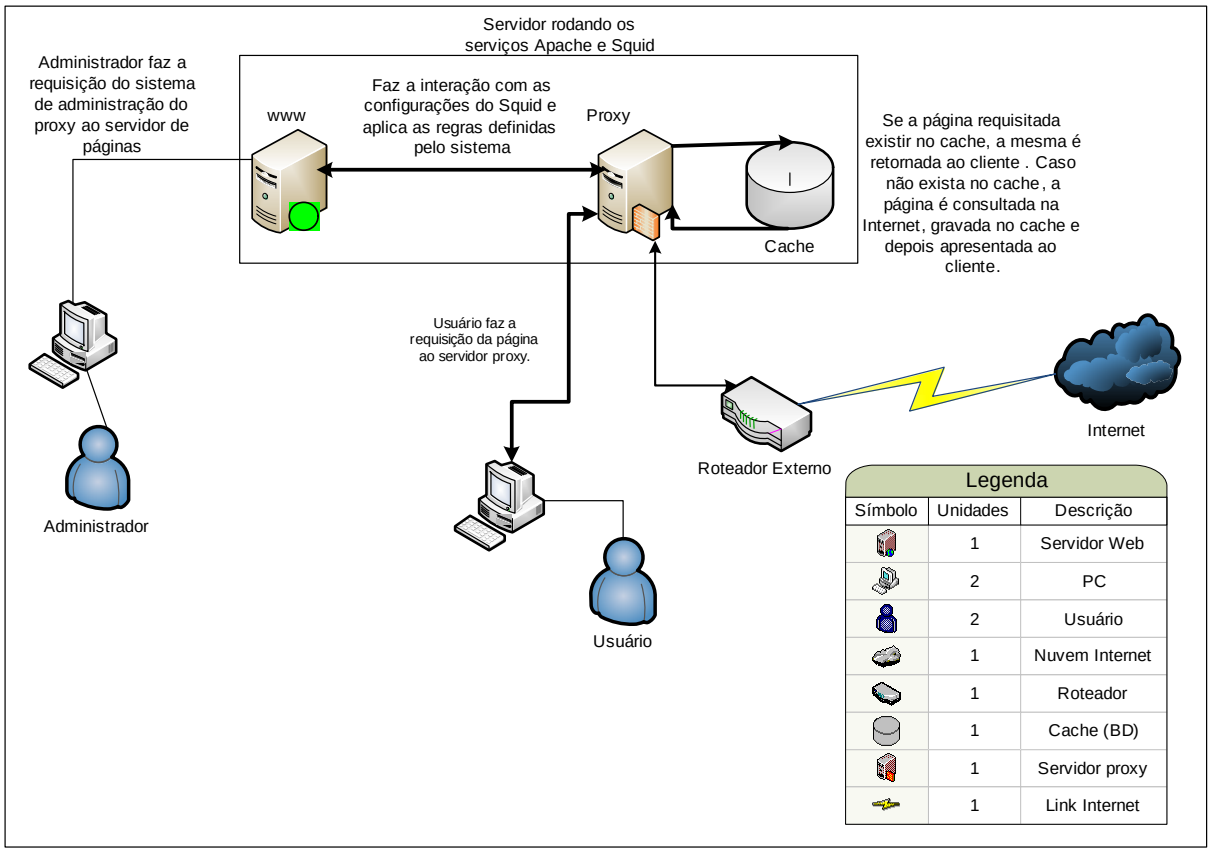

Figura 7 – Funcionamento da ferramenta e requisições ao *proxy*

3.3.2 Operacionalidade da implementação

Conforme descrito no diagrama de casos de usos, a ferramenta *web* para administração do servidor *proxy* Squid é uma página de Internet que permite a implementação da política de controle de acesso, de forma implícita pelo administrador da rede.

A ferramenta *web*, possibilita que um administrador menos experiente, possa configurar o Squid, sem interação alguma com o seu arquivo de configuração em modo texto, isso tudo através de uma linguagem mais comum e sem temos muito técnicos. Permite o cadastro, consulta e alteração de usuários em seus grupos, estes definidos pela ferramenta, o bloqueio de *downloads* com filtros por extensões de arquivos, o bloqueio por palavras proibidas, a liberação ou bloqueio de páginas da Internet, a verificação e configuração básica dos relatórios de acesso à Internet, a verificação em tempo real do *log* de acesso à Internet, reiniciar o Squid e configurar a ferramenta *web*, conforme a distribuição do GNU/Linux utilizada.

Abaixo seguem as telas principais da ferramenta *web*.

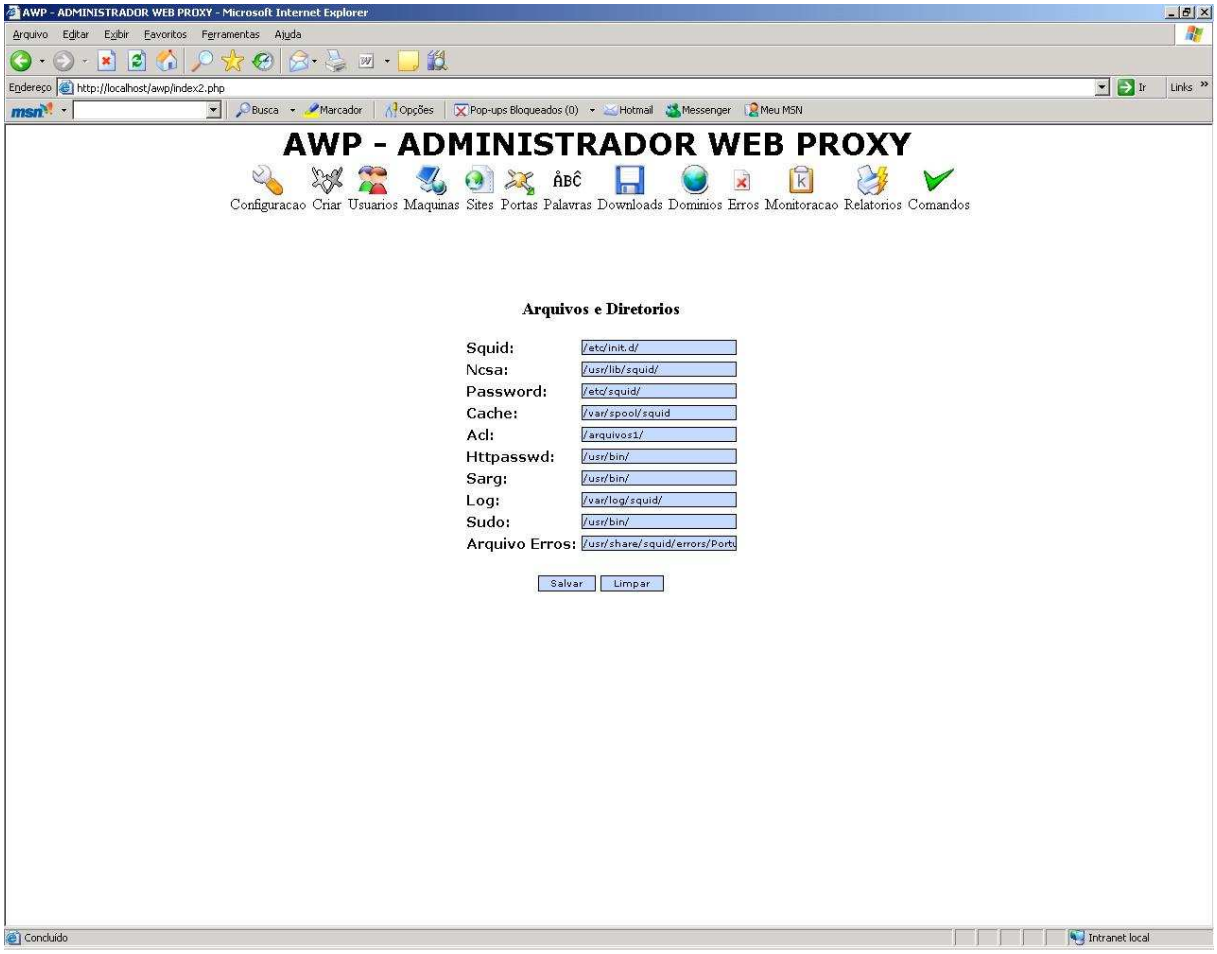

Figura 8 – Tela de parâmetros de arquivos e diretórios

Na Figura 8, é apresentada a tela de configuração dos parâmetros da ferramenta, serão informados os caminhos absolutos dos diretórios onde os arquivos utilizados pela ferramenta se encontram.

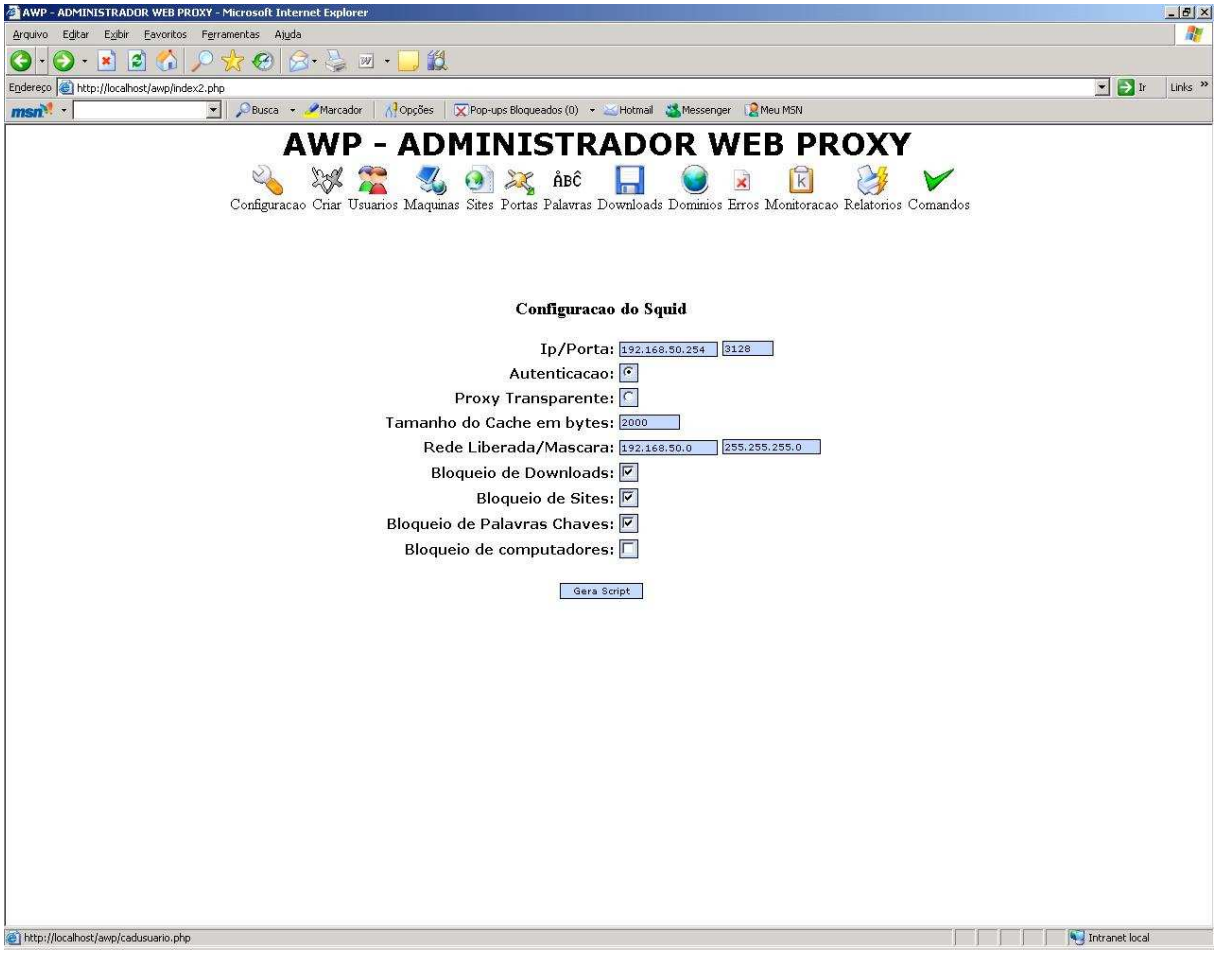

Figura 9 – Tela de configuração do Squid

Na Figura 9, é apresentada a tela de configuração do Squid, serão inseridas as informações que serão gravadas no arquivo `squid.conf` que é o arquivo de configuração do servidor.

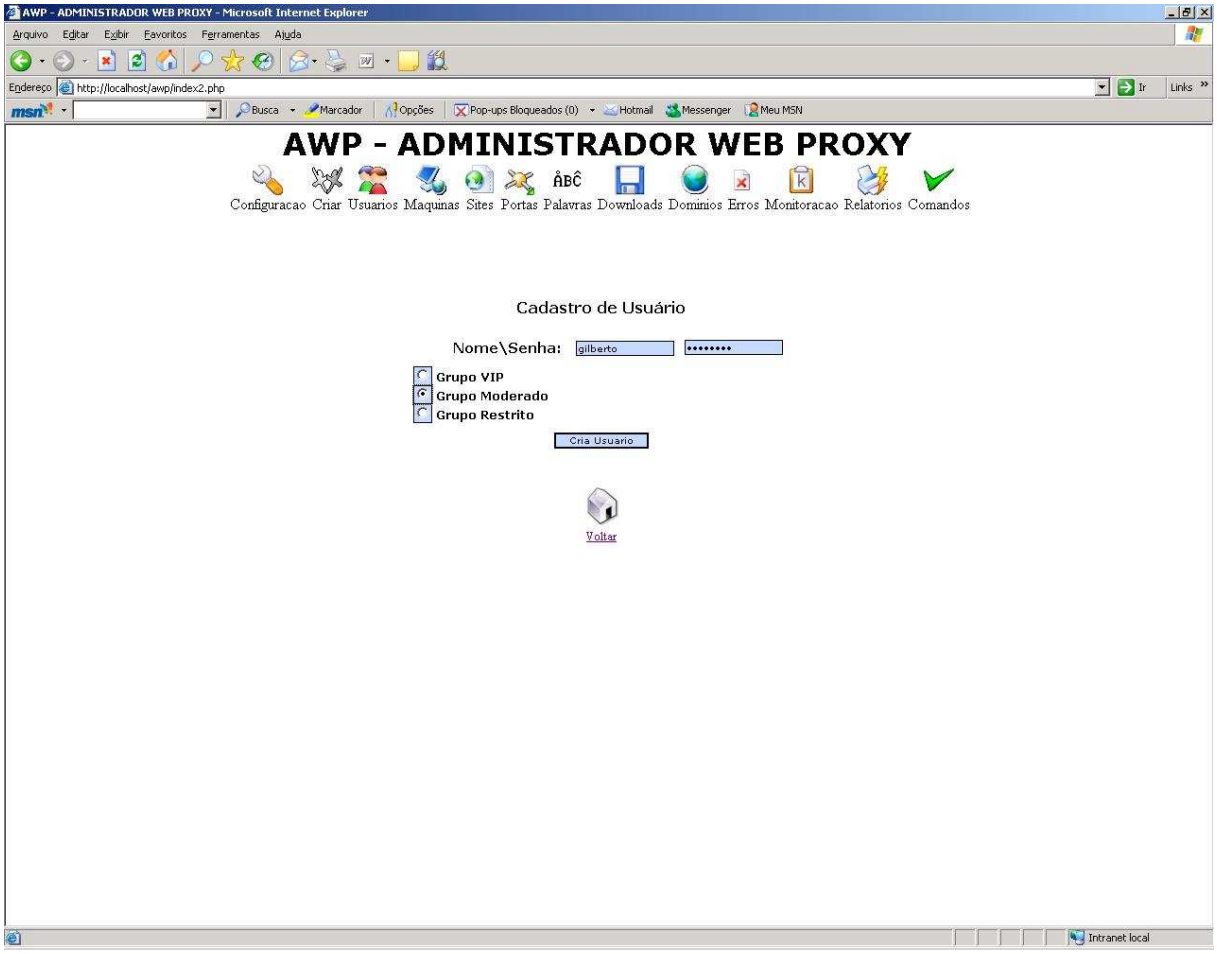

Figura 10 – Tela de cadastro de usuários

Na Figura 10, é apresentada a tela de cadastro de usuários, será inserido o nome e senha do usuário que se deseja cadastrar e selecionar o grupo ao qual esse usuário deverá estar associado.

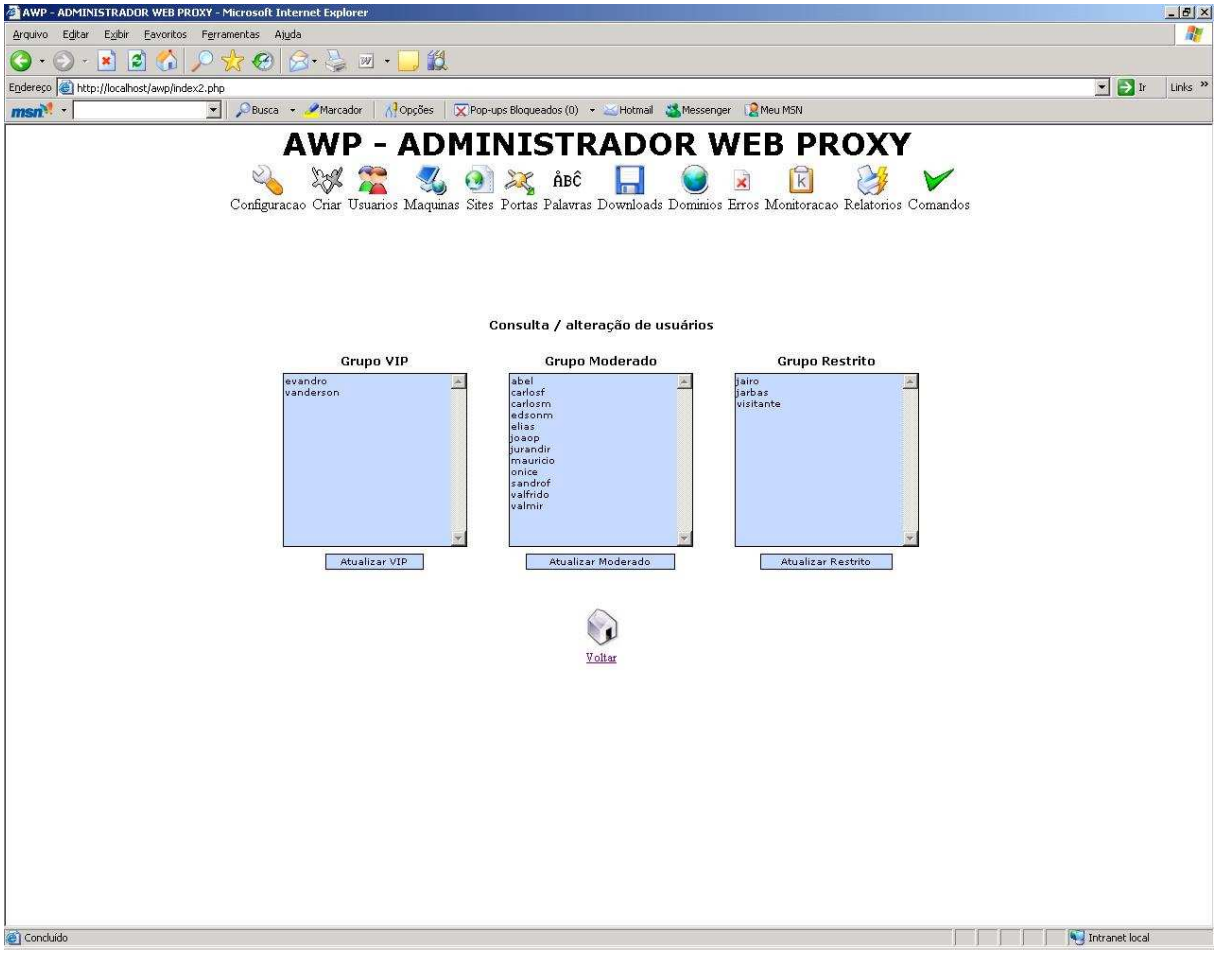

Figura 11 – Tela de consulta / alteração de usuários nos grupos

Na Figura 11, é apresentada a tela de consulta / alteração de usuários, permite verificar em que grupo determinado usuário está cadastrado e se for necessário pode-se alterar o mesmo de grupo. Permite também excluir esse usuário, para que o mesmo não tenha mais acesso à Internet.

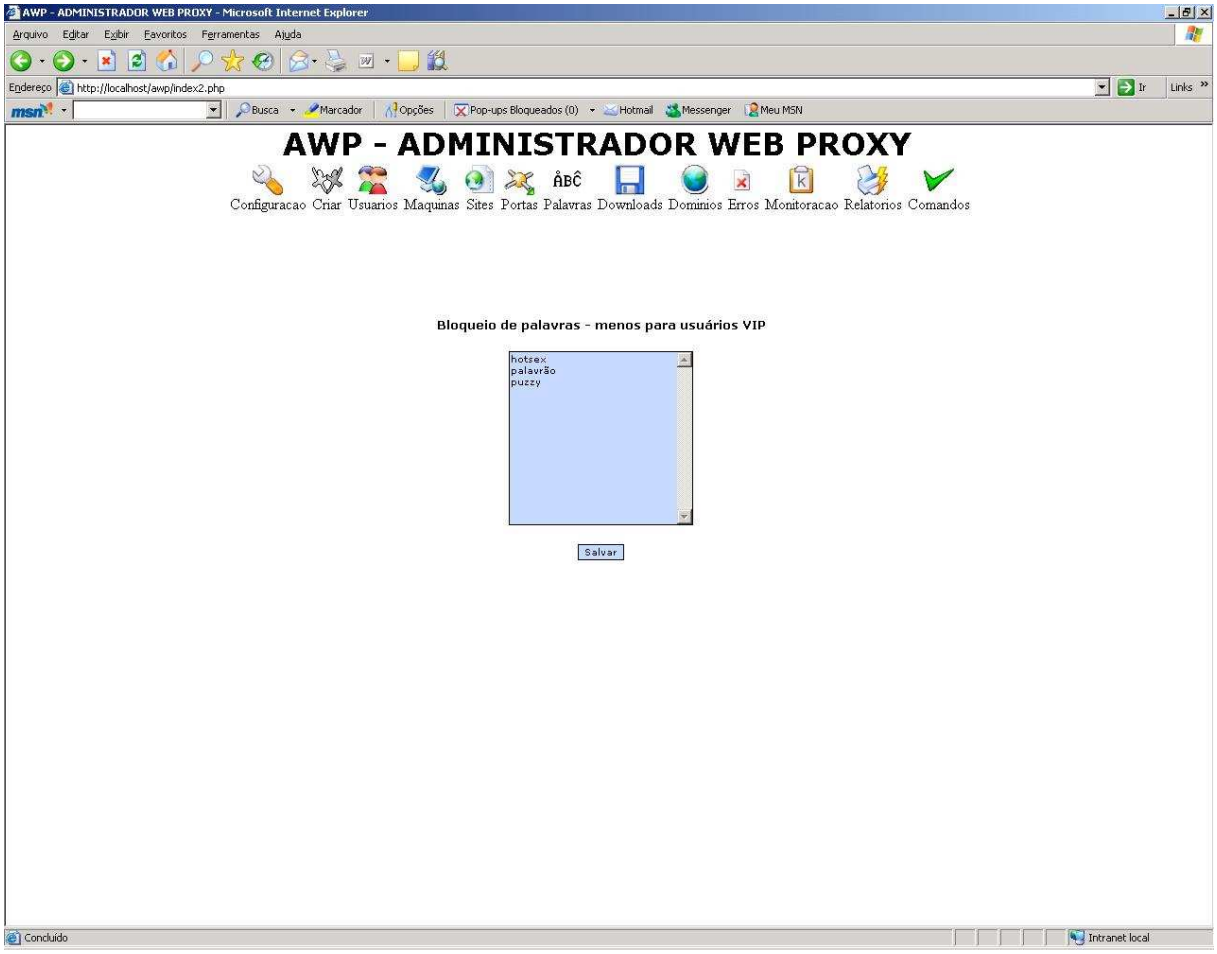

Figura 12 – Tela de bloqueio de palavras

Na Figura 12, é apresentada a tela de bloqueio de palavras, permite consultar as palavras que estão sendo bloqueadas pelo servidor *proxy* e pode-se excluir ou inserir novas palavras.

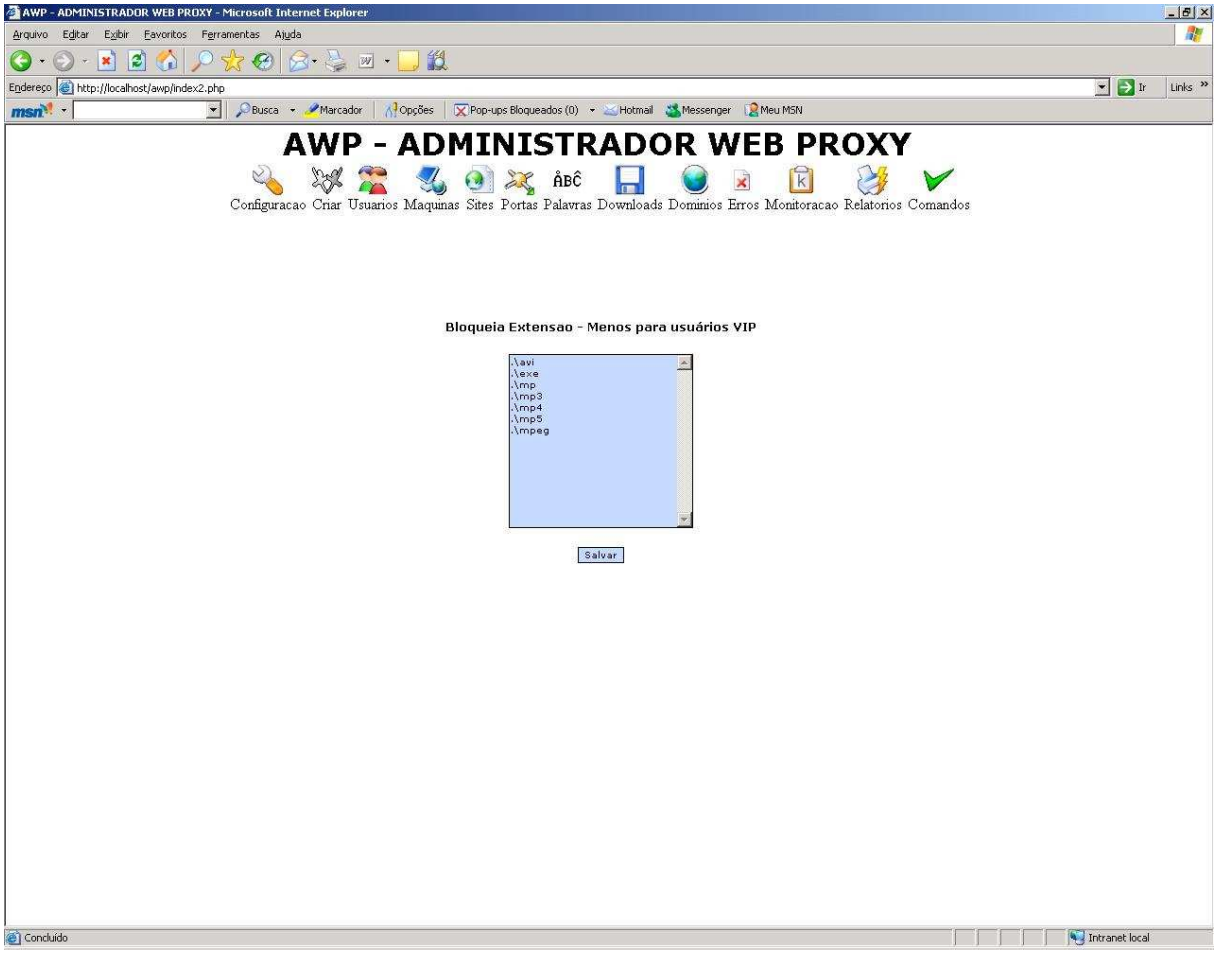

Figura 13 – Tela de bloqueio de *downloads*

Na Figura 13, é apresentada a tela de bloqueio de *downloads*, permite consultar as extensões de arquivos que estão sendo bloqueados para *download* pelo servidor *proxy* e pode-se excluir ou inserir novas extensões.

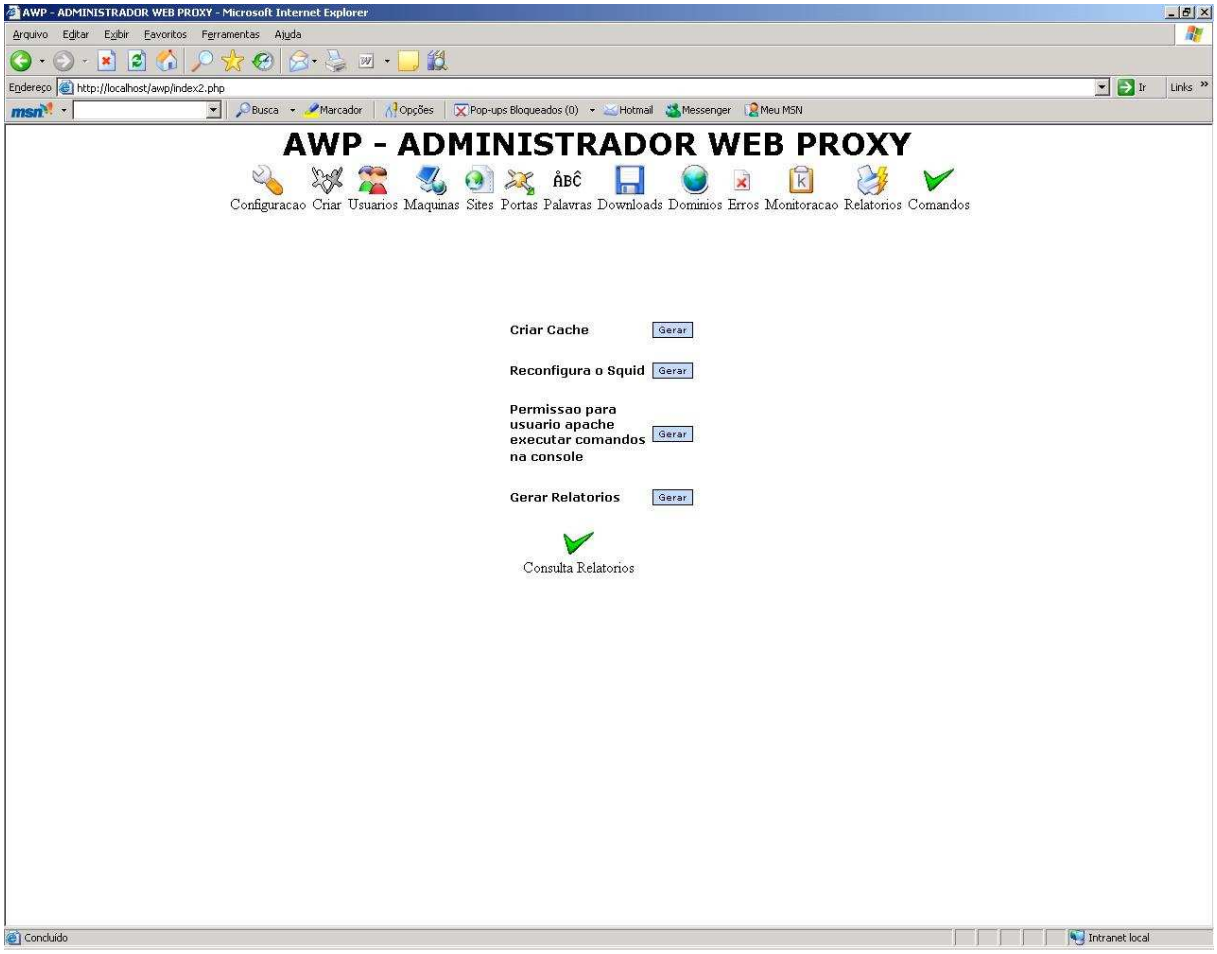

Figura 14 – Tela de comandos

Na Figura 14, é apresentada a tela de comandos, permite ao administrador reiniciar o Squid, criar o *cache* do *proxy*, configura a permissão para o usuário apache executar os comandos na *console* com direitos de super usuário, gerar o relatório de acesso à Internet e consultar os relatórios já gerados anteriormente.

3.4 RESULTADOS E DISCUSSÃO

Nos testes feitos em laboratório a ferramenta *web* mostrou-se eficiente na interação com as configurações do Squid, geração de relatórios, manipulação de usuários e na personalização das páginas de erros que são mostradas aos usuários.

Os principais testes realizados e seus resultados estão demonstrados no Quadro 15.

Descrição do teste	**Resultado obtido**
Validação das mensagens de erros da ferramenta	Foram verificadas todas as mensagens de erros e possíveis exceções do programa. Todas as mensagens foram revistas e reformuladas conforme a necessidade.
Administrador cadastrou usuários em grupo errado	Foi feita a alteração do usuário para outro grupo de acesso.
Administrador cadastrou palavra proibida	Foi feito teste com usuários dos grupos de acesso restrito e moderado, a requisição a página solicitada foi bloqueada. No caso do grupo VIP, foi liberada.
Administrador cadastrou extensão de *download* proibido	Foi feito teste com usuários dos grupos de acesso restrito e moderado, o *download* do arquivo com a extensão cadastrada, foi bloqueada. No caso do grupo VIP, foi liberado.
Administrador gerou o relatório de acesso a Internet	Foi feita a consulta do relatório de acesso à Internet, que apresentou as páginas acessadas pelos usuários.
Administrador alterou configuração do Squid	Com a alteração da configuração foi feito teste de acesso à páginas da Internet, e como a configuração do Squid estava prevendo outra rede local, não navegou.
Administrador reiniciou o Squid	Foi feito o acompanhamento do processo de reinicio do Squid na console do servidor, pelo *log* de informação do servidor. Reinicio foi feito com sucesso.

Quadro 15 – Testes com a ferramenta e resultados obtidos

Uma vantagem importante observada nessa ferramenta *web* em comparação com seu correlato chamado protótipo de ferramenta *web* para gerenciamento de *firewall* (BORCHEID, 2005), é que a ferramenta *web* implementada aqui não é baseada em um filtro de pacotes, mas sim em um filtro de conteúdo.

Com relação ao seu correlato chamado Webmin (ZAGO, 2007), a ferramenta *web* desenvolvida, oferece uma linguagem mais acessível e menos técnica, além de ser uma ferramenta bem mais específica que o correlato.

O correlato chamado SARG (ORSO, 2006), oferece somente a funcionalidade de gerar os relatórios de acesso do *proxy*, sendo assim, foi utilizado para essa função.

4 CONCLUSÕES

Este trabalho teve por objetivo o desenvolvimento de uma ferramenta *web* que auxilie na administração do servidor *proxy* Squid através da *web*.

Foi desenvolvido uma ferramenta *web* voltada para a administração do servidor Squid, de fácil compreensão mesmo para administradores que têm somente uma noção superficial do que é o Squid, possibilitando fazer a configuração do mesmo por meio de simples seleções e preenchimento de alguns formulários, conforme necessidade. Possibilitando ainda a utilização de grupos de acesso à Internet para os usuários, levando em consideração a manipulação das políticas e controles de acesso a *sites*, sendo algo totalmente diferente das ferramentas existentes, que manipulam apenas os serviços mais utilizados de forma mais intuitiva para usuários com um conhecimento mais avançado da plataforma de código aberto GNU/Linux, como é o caso do Webmin.

Sendo assim, o Webmin não é muito difundido porque utiliza uma linguagem muito técnica (PCMASTER, 2006). A ferramenta desenvolvida por Borscheid (2005) manipula somente a função de filtro de pacotes, o que não permite bloquear acessos à *sites* da Internet, *downloads* de arquivos, gerência de usuários, saber especificamente onde os usuários navegaram e não consegue bloquear aplicações, como por exemplo, algum jogo diretamente em um *site* da Internet.

Conforme Orso (2006), a ferramenta *chpasswd* somente manipula as senhas de acesso dos usuários do Squid, trabalhando em conjunto com o utilitário de *console htpasswd*. A alteração é muito superficial comparada com as funcionalidades que o Squid oferece, como por exemplo, alterar alguma política de acesso a *sites*.

O SARG é uma ferramenta muito difundida para fazer a integração com o Squid, por se tratar do sistema que faz toda a análise de *logs* de acesso e que transforma estas informações em uma linguagem de fácil entendimento para o usuário, através de uma página da Internet.

Com o intuito de facilitar as atividades do administrador de rede, principalmente para os iniciantes nas configurações do servidor Squid, a proposta desta ferramenta é aumentar a confiabilidade para evitar que alguma alteração seja feita de forma indevida diretamente no arquivo de configuração do Squid, principalmente por ser feito no editor de texto nativo do GNU/Linux, o *vi*, aumentando a eficiência da manutenção do servidor *proxy* Squid, pois o administrador saberá exatamente o que está alterando e como esta mudança vai refletir no

sistema.

4.1 EXTENSÕES

As sugestões para extensão desse trabalho são:

a) implementação de sessão na ferramenta, pois atualmente a ferramenta não tem o controle da sessão da página;
b) implementação que permita utilizar as demais configurações do Squid, que são muitas, pois a ferramenta suporta somente as principais;
c) implementação que permita ao administrador de rede criar grupos de acesso conforme sua necessidade;
d) implementação que permita configurar perfis de acesso de usuários na ferramenta, isso para redes maiores, onde não existe somente um administrador de rede;
e) implementação que permita a configuração avançada do SARG.

Todas as sugestões são visando a utilização do servidor *proxy* Squid, que é o mais utilizado pelas empresas atualmente.

REFERÊNCIAS BIBLIOGRÁFICAS

APACHE HTTP SERVER. In: WIKIPEDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2007. Disponível em: < http://en.wikipedia.org/wiki/Apache_server>. Acesso em: 21 abr. 2007.

BAROS, E. B. **Configurando um Squid "ninja"**. [S.l.], [2006?]. Disponível em: <http://www.linuxman.pro.br/squid/>. Acesso em: 02 abr. 2006.

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML.** Rio de Janeiro: Campus, 2002.

BORSCHEID, R. M. **Protótipo de aplicação web para gerenciamento de firewall em Linux**. 2005. 52 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CAMPOS, A. L. N. **Sistema de segurança da informação:** controlando os riscos. Florianópolis: Visual Books, 2006.

CHADD, A. et al. **Squid web proxy cache**. [S.l.], [2006?]. Disponível em: <http://www.squid-cache.org>. Acesso em: 30 mar. 2006.

CONTROLE DE ACESSO. In: WIKIPEDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2007. Disponível em: <http://pt.wikipedia.org/wiki/Controle_de_acesso>. Acesso em: 24 abr. 2007.

EQUIPE CONECTIVA. **Segurança de redes:** firewall. [Curitiba]: Conectiva S.A., 2001.

JESUS, D. C. S. de et al. **Implantando WCCP na hierarquia de proxies da RNP**. [Rio de Janeiro], [2001]. Disponível em: < http://www.rnp.br/newsgen/0103/wccp.html>. Acesso em: 21 abr. 2007.

LIMA, M. M. de A. E. **Introdução a gerenciamento de redes TCP/IP**. [Rio de Janeiro], [1997]. Disponível em: <http://www.rnp.br/newsgen/9708/n3-2.html>. Acesso em: 16 abr. 2007.

MARCELO, A. **Squid:** configurando o proxy para Linux. 4. ed. Rio de Janeiro: Brasport, 2005.

NEMETH, E. et al. **Manual do administrador do sistema Unix**. 3. ed. Tradução Edson Furmankiewicz. Porto Alegre: Bookman, 2002.

ORSO, P. **SARG:** Squid Analysis Report Generator. [S.l.], [2006?]. Disponível em: <http://sarg.sourceforge.net>. Acesso em: 29 mar. 2006.

PALMA, L.; PRATES, R. **TCP/IP:** guia de consulta rápida**.** São Paulo: Novatec, 2000.

PCMASTER. **Administrando o Linux pela internet com o Webmin**. [S.l.], [2005?]. Disponível em: <http://www.linuxnarede.com.br/tutoriais/post_art/fullnews.php?id=view&f_act=fullnews&f_id=89/>. Acesso em: 22 maio 2006.

PÉRICAS, F. A. **Redes de computadores:** conceitos e a arquitetura Internet. Blumenau: Edifurb, 2003.

PROXY. In: WIKIPEDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2007. Disponível em: < http://pt.wikipedia.org/wiki/Proxy>. Acesso em: 16 abr. 2007.

SAUVÉ, J. P. **Gerência de redes de computadores.** [Campina Grande]: Departamento de sistemas e computação da universidade federal de Campina Grande – Paraíba, [2002?]. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/2002.1/gr/>. Acesso em: 16 abr. 2007.

SILVA, E. dos S. da. **Extensão do modelo de restrições do RBAC para suportar obrigações do modelo ABC**. 2004. 90 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba.

VESPERMAN, J. **Autenticação e o squid**. [S.l.], [2001]. Disponível em: < http://br.geocities.com/cesarakg/AuthenticationAndSquid.html>. Acesso em: 21 abr. 2007.

WATANABE, C. S. **Introdução ao cache de web**. [Rio de Janeiro], [2000]. Disponível em: <http://www.rnp.br/newsgen/0003/cache.html>. Acesso em: 18 abr. 2007.

ZAGO, A. F. **FAQ:** dicas e indicações de tutoriais sobre webmin, configurador em ambiente gráfico. [S.l.], [2007?]. Disponível em: <http://www.zago.eti.br/webmin.txt>. Acesso em: 18 abr. 2007.