Gustavo Hendrigo Marcon

PhpDansAdmin, uma ferramenta web para administração do DansGuardian

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Orientador
Prof. MSc. Herlon Ayres Camargo

Lavras
Minas Gerais - Brasil
2010

Gustavo Hendrigo Marcon

PhpDansAdmin, uma ferramenta web para administração do DansGuardian

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Aprovada em *Dezembro de 2010*

Prof. Sanderson L. Gonzaga de Oliveira

Prof.Joaquim Quinteiro Uchôa

Prof. MSc. Herlon Ayres Camargo
(Orientador)

Lavras
Minas Gerais - Brasil
2010

As pessoas da minha vida que me incentivam: Meus pais, minha tia Maria Ap. Marcon Gimenes e minha namorada Patricia O. Peixoto.

Agradecimentos

Agradeço a meu amigo Bruno P. Gonçalves (Scorninpc) por me ajudar no desenvolvimento PHP, ao meu orientador MSc. Herlon Ayres Camargo pelas orientações e aos demais professores do curso Administração em Redes Linux, que fizeram valer a pena todo esse esforço.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

O objetivo deste trabalho é apresentar a construção de uma ferramenta web que facilite as operações de gerenciamento de acessos à internet em servidores Proxies que utilizem o DansGuardian como filtro. Através desta interface, denominada **PhpDansAdmin**, o administrador digita ou seleciona as opções desejadas por meio de um navegador comum. Desenvolvida em PHP, auxilia nas tarefas de inclusão e exclusão de grupos (perfis), modificações dos direitos de acessos, atribuições de grupos e configurações de sistema no DansGuardian.

Palavras-Chave: Proxy, Squid, Dansguardian, web PHP, internet.

Capítulo 1

Introdução

Com o rápido crescimento da internet no meio corporativo, e atrelado a ele o aumento dos incidentes na segurança da informação, é cada vez maior a preocupação com a política de segurança de rede. "A política de segurança da empresa vai deixar claro que áreas devem ter acesso restrito e quais podem ser liberadas sem problemas em uma rede privada. É ela que define quais itens precisam ser preservados, bem como quais pessoas terão acesso a determinados recursos" (UCHÔA, 2005).

Em se tratando de acesso à internet, a forma mais comum de controle de acessos é através do *proxy*. Segundo (UCHÔA; SICA; SIMEONE, 2003), *proxy* é um programa que se situa entre a rede local e o mundo externo (internet), realizando o controle na comunicação entre os dois lados. Dessa forma, o cliente não tem acesso direto à internet, o que constitui uma vantagem do ponto de vista de segurança.

1.1 Motivação

Diante da necessidade de corporações gerenciarem a política de acessos à internet, surgem demandas de ferramentas que melhorem a administração dos servidores. A origem do desenvolvimento dessa ferramenta ocorreu quando o autor deste trabalho executava várias alterações diárias dos tipos de perfis de acessos à internet em uma rede que utiliza um servidor *proxy/cache* com o DansGuardian como filtro.

O modelo adotado de desenvolvimento utiliza ferramentas comuns como o OpenSSH e pode ser também aproveitado para criar novas ferramentas que efetuem ajustes em outros tipos de servidor, que tenham por ventura a necessidade de administração e não apresentam opções que facilitem o seu gerenciamento.

1.2 Objetivos

1.2.1 Objetivo Geral

O presente estudo apresenta o desenvolvimento de uma aplicação *web* para o gerenciamento do DansGuardian.

1.2.2 Objetivos Específicos

Por intermédio de um navegador *web*, a aplicação desenvolvida permitirá efetuar as seguintes ações no DansGuardian:

- criar e excluir grupos;
- associar usuários aos grupos criados;
- personalizar os perfis de acessos à internet dos grupos;
- atualizar a lista negra utilizada;
- executar alterações nos arquivos de configurações.

1.3 Metodologia

No decorrer deste trabalho serão utilizadas, basicamente, duas estratégias metodológicas, sendo a primeira delas a pesquisa bibliográfica, utilizada para apresentar toda a parte conceitual do documento, compreendida no capítulo 2 com a abordagens de diversos autores sobre os temas tratados.

A segunda estratégia metodológica utilizada é o desenvolvimento de aplicações diversas, apresentadas nos capítulos 3 e 5, abordando o trabalho de desenvolvimento e teste da solução, a partir de um caso real.

1.4 Organização do Trabalho

O trabalho está dividido em seis capítulos, sendo que o embasamento de estudo e o desenvolvimento estão restritos entre o capítulo 2 e o 5.

O capítulo 2, intitulado de Revisão da Literatura, aborda assuntos relacionados aos softwares Squid, DansGuardian, Apache e OpenSSH, que são necessários para o funcionamento do PhpDansAdmin, e exemplos de ferramentas *web* desenvolvidas em PHP utilizadas na administração de serviços.

O capítulo 3 apresenta o desenvolvimento da ferramenta e os mecanismos utilizados para o gerenciamento do DansGuardian. O capítulo 4 apresenta a execução da instalação da aplicação. No capítulo 5 é executado um teste em que é demonstrado uma utilização prática de uso da ferramenta. O capítulo 6 traz a conclusão com base em resultados obtidos na utilização da ferramenta.

Capítulo 2

Revisão da Literatura

Neste capítulo serão apresentados os softwares e configurações utilizados no ambiente e são indispensáveis para o cumprimento dos objetivos deste trabalho. No final, há alguns exemplos de ferramentas de gerenciamento que também foram desenvolvidas na linguagem PHP, assim como o PhpDansAdmin.

2.1 Servidor proxy

Com o crescente avanço no uso da internet, a infraestrutura das empresas que atuam como provedores desse serviço não conseguiu acompanhar o rápido crescimento da rede mundial, resultando em lentidão no acesso às informações que trafegam pela rede.

Para amenizar o problema, foram criados os servidores de *proxy/cache*. Quando um terminal da rede interna acessa uma página, o servidor *proxy/cache* salva uma cópia com data e horário em que a página foi criada. Quando outro terminal da rede realiza a requisição para a mesma página o servidor compara a data e o horário da página salva com a página da internet, se a data e o horário forem iguais, o servidor *proxy* não precisa realizar o *download* de toda a página da internet novamente (MARCELO, 2006).

Agregada à função de *cache*, foi adicionado o controle de acesso à internet, que é outra funcionalidade bastante utilizada, uma vez que o cliente não tem acesso direto à internet e deve fazer requisição ao servidor *proxy* para acessá-la. Dentre os diversos *proxies* disponíveis no mercado, um dos mais utilizados em todo o mundo

é o Squid, que é um software livre e possui versões para diferentes distribuições Linux. Além disso, destaca-se pelo excelente desempenho e grande flexibilidade de configuração.

2.2 Squid

O Squid é um poderoso servidor de *proxy/cache* e oferece suporte para os principais sistemas operacionais baseados em UNIX, dentre eles FreeBSD, OpenBSD, SunOS, HP-UX, AIX, e atualmente acompanha as principais distribuições de Linux. Licenciado nos termos da GNU *General Public License* (GPL) da *Free Software Foundation*. O Squid é largamente utilizado para compartilhamento de acesso à *web*, possuindo características que permitem ainda trabalhar com outros objetivos, como melhoria da *performance* e controle de acesso (COSA, 2007).

O Squid trabalha como intermediário entre cliente e servidor, ou seja, recebe as requisições e repassa aos servidores. Essa característica pode gerar uma confusão com o *Network Address Translation* (NAT) porém o *proxy*, diferente do NAT, baseia-se na aplicação. Por trabalhar com uma aplicação específica, permite controle maior sobre vários serviços, tais como aplicações que podem usar qualquer porta, já que o Squid trabalha em portas e aplicações predefinidas (RUFINO, 2002).

O Squid foi feito para trabalhar com o protocolo HTTP, mas com a expansão da internet, outros serviços além do HTTP estão sendo utilizado pelo *proxy*. Alguns dos protocolos que o Squid suporta são: HTTP, FTP, SSL, ICP, HTCP, CARP, WCCP, SNMP e DNS. Mas há protocolos como o POP e o SMTP, por exemplo, que não trabalham com os citados anteriormente, sendo necessária a utilização de outros métodos de acesso à internet (TRIGO, 2007).

Há várias soluções encontradas no mercado para o uso do *proxy*, na implementação deste trabalho foi utilizado o Squid que é gratuito. Com ele é possível construir um servidor com as principais características que outras soluções proprietárias fariam.

Dentre as funcionalidades no uso do Squid, destacam-se:

- menor consumo da banda na ligação à internet;
- redução no tempo para recarregar páginas *web*;
- proteção aos *hosts* da rede interna;

- coleta de dados estatísticos sobre o tráfego da *web* na rede;
- impedimento para que os usuários visitem sites inapropriados (no trabalho ou na escola, por exemplo);
- garantia de que somente usuários autorizados possam navegar na internet;
- redução da carga no *link* de internet.

A explanação das funcionalidades do Squid deixa bem explícita seus principais propósitos: o ganho de performance e a segurança. A Figura 2.1 mostra o esquema de funcionamento de um sistema de acesso à internet baseado em *proxy*.

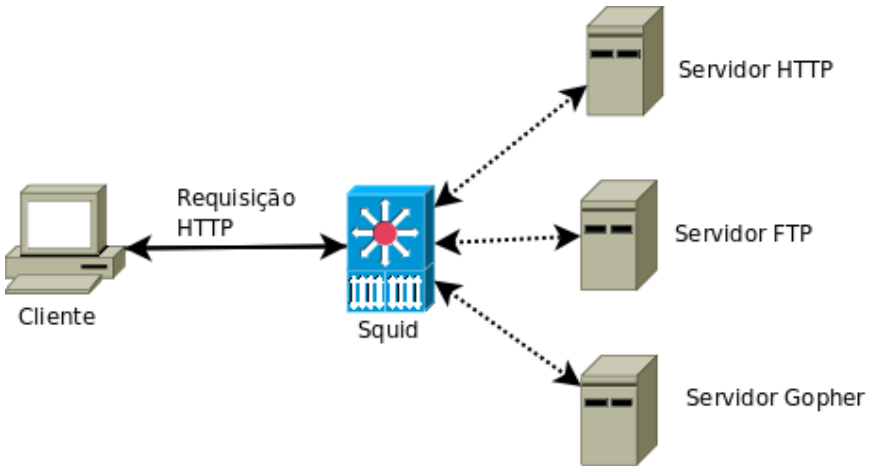

Figura 2.1: Esquema de funcionamento de sistema de acesso à internet baseado em *proxy*. (WESSELS, 2004).

2.2.1 Requisitos de Hardware e Software

Os requisitos de hardware para o Squid geralmente são modestos, a memória é muitas vezes o recurso mais importante para seu melhor desempenho e a falta dela é um dos sérios motivos para sua degradação.

Outro fator que merece ser destacado é o espaço em disco, uma vez que, quanto maior for o espaço, maior será o *cache*. CPUs potentes não são essenciais para um bom desempenho, porque o Squid utiliza pequena quantidade de processamento para cada resposta.

A regra geral diz que há uma relação entre o espaço em disco e requisitos de memória; em média são necessários 32 MB de memória RAM para cada 1 GB de *cache* em disco, mas isso pode variar dependendo de vários fatores, tais como: arquitetura de CPU (32 ou 64 bits), número de usuários simultâneos e outras características específicas (WESSELS, 2004).

2.2.2 Instalação

Segundo (SIQUEIRA, 2009), várias distribuições Linux possuem um pacote Squid, que facilita a instalação e mesmo que seja a partir do código fonte é um fator corriqueiro.

O processo de instalação do Squid é simples. Neste trabalho será utilizada a versão 2.7 em um servidor com sistema operacional GNU/Linux Debian 5.0. Para instalar o serviço deve-se digitar o comando `apt` com o pacote requerido, que pode ser visto na Figura 2.2.

```
# apt-get install squid
```

Figura 2.2: Instalação do Squid no Debian.

Ao término desse passo, o Squid estará instalado: o diretório com os arquivos de configurações encontra-se em `/etc/squid`.

2.2.3 Configurações

Alterando-se poucas opções no arquivo `squid.conf` é possível efetuar uma configuração básica para que trabalhe de forma transparente ou autenticada. Para melhor controle de acesso, a configuração mais indicada é a autenticada. Nesse tipo de configuração é possível realizar o controle por usuários, uma vez que poderão utilizar qualquer terminal da rede e terão acessos garantidos ou bloqueados de acordo com os controles do filtro, solicitando apenas um login e senha antes do início da navegação.

A autenticação, segundo (SQUID, 2010), poderá ser feita pelo próprio Squid ou outro software, por meio de mecanismos de autenticação suportados, entre os quais estão:

- ntlm_auth — utilizado por clientes windows que ao realizarem login na rede passam dados ao Squid, que bloqueia ou permite o acesso;

- external_acl — passa um parâmetro de ERRO ou OK a qualquer mecanismo de autenticação externa;

- basic_auth — utiliza o algoritmo base64 para trafegar senhas através do protocolo HTTP;

- digest_auth — utiliza um esquema de perguntas e respostas baseado no algoritmo HMAC, para trafegar senhas no protocolo HTTP.

Um exemplo completo com o método *basic* de autenticação e que atenda às necessidades da ferramenta proposta neste trabalho encontra-se no Apêndice A.1. Após efetuar as configurações básicas, editando o arquivo `/etc/squid.conf` é possível verificar se o processo está sendo executado na porta 127.0.0.1:3128. A melhor maneira de fazê-lo na prática é com o uso do comando `netstat` com a opção `-ant`.

2.3 DansGuardian

O DansGuardian[1] é uma opção de filtro de conteúdo desenvolvido para trabalhar em conjunto com o Squid. Utilizando um filtro adaptativo, esta ferramenta avalia a página e delibera se ela é uma página imprópria com base no conteúdo, empregando um conjunto de regras configuráveis. Este filtro inclui um conjunto de regras prontas, que contêm palavras, frases e tipos de arquivo frequentemente usados em páginas impróprias, além de uma lista de páginas conhecidas. A filtragem é feita cruzando-se todas essas informações.

Atualmente é suportado em ambientes Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X, HP-UX, e Solaris e filtra o conteúdo de páginas baseadas em vários métodos, incluindo correspondência de frase, rótulo de imagens e filtragem de URL (DansGuardian Organization, 2009).

Projetado para ser completamente flexível, permite a adaptação da filtragem de acordo com as necessidades do administrador, para que se obtenham os melhores resultados.

[1] `http://www.dansguardian.org/`.

Trata-se de um produto "semicomercial", possui o código aberto e é gratuito para uso pessoal ou para qualquer fim que não vise à comercialização. Entretanto, o produto deverá ser pago caso essa finalidade não seja atendida, tornando o uso comercial (DansGuardian Organization, 2009).

Na filtragem de frases de conteúdo, o programa verifica páginas que contêm palavrões, frases muitas vezes associadas a pornografia e outros conteúdos indesejáveis. Permite bloquear ou limitar *uploads* na internet, e o filtro de URL e domínio é capaz de lidar com listas enormes e é significativamente mais rápido que outros filtros semelhantes. Seus arquivos de *logs* é produzido em um formato de fácil de leitura e também pode ser gerado no mesmo formato do Squid.

O DansGuardian pode ser personalizado conforme o administrador desejar, podendo ser severo ou desobstrutivo sobre o que é filtrado.

O DansGuardian conta com muitos recursos. Dentre as principais características destacam-se as seguintes:

- para uso comercial é significativamente mais barato que outras soluções como, por exemplo, *web*sense[2];
- pode filtrar as páginas obscenas (sexual, racial, violência, etc.);
- utiliza uma avançado sistema de ponderação de frases sobre ou sob bloqueio;
- é possível filtrar sites usando os rótulos de arquivos de imagens (PICS);
- permite filtrar tipos e extensão de arquivo;
- permite filtrar URLs, inclusive URLs com expressões regulares;
- filtragem de URL compatível para o uso da *blackList*;
- filtragem de URL capaz de filtrar requisições https;
- permite trabalhar com listas de exceções;
- permite bloquear URLs baseadas em IP;

[2]A *web*sense é lider mundial em soluções URL *filtering*, com base de dados de sites de internet, classificados por categoria e com atualização diária. O funcionamento conjunto com um firewall (por exemplo) permite dar ou impedir acesso a certas categorias de sites para o conjunto dos utilizadores ou para certos grupos, configurando diversas políticas de acesso. Site oficial: http://www.*web*sense.com

- é capaz de bloquear sites quando os usuários tentam acessar o endereço pelo IP;
- produz *logs* em um formato muito legível;
- opcionalmente produz *logs* em formato CSV[3] para fácil importação em bancos de dados;
- é capaz de registrar o nome de usuário usando autenticação de *proxy base*;
- pode bloquear ou limitar *upload* na internet (por exemplo, anexos no Hotmail);
- utiliza um algoritmo inteligente para coincidir frases em páginas da *web* misturadas em códigos HTML e espaços em branco;
- filtragem de URL significativamente mais rápida que o SquidGuard[4];
- funciona perfeitamente em conjunto com o Squid;
- suporta X-Forwarded-For no Squid, que permite repassar os IPs clientes para o DansGuardian.

2.3.1 Requisitos de *Hardware*

O DansGuardian 2.9 exige ao menos 300Mhz de processamento, 64 MB de memória RAM por grupo de cinquenta usuários e no mínimo 4 Gb de disco rígido. Se for utilizado com conexões rápidas com a internet (como 512 Kbps ou mais) será necessário alocar mais processamento (DansGuardian Organization, 2009).

2.3.2 Configurações

Para o funcionamento do DansGuardian, é preciso que o Squid já esteja instalado e configurado. O DansGuardian adiciona políticas de acessos, mas deixa

[3] O CSV é um implementação particular de arquivos de texto separados por um delimitador, que usa a vírgula e a quebra de linha para separar os valores.

[4] SquidGuard é um plugin para Squid que facilita ao administrador a manutenção e aplicação de regras de controle sobre o acesso dos usuários, além de implementar alguns controles não existentes no próprio Squid. Possui um arquivo de configuração à parte do Squid. Uma das vantagens de se usar o SquidGuard é o fato de usar um banco de dados para acessar *blacklists*, o que o torna muito mais rápido que o bloqueio usando *blacklists* em arquivos de texto. Site oficial: http://www.squidguard.org/

que o próprio Squid faça o acesso à internet, *cache* e autenticação. Para o DansGuardian não importa como está sendo feita a autenticação, o programa só recebe os dados de *login* repassados pelo Squid. O principal arquivo de configuração é o `dansguardian.conf` do diretório `/etc/dansguardian/`. Na edição, a linha `UNCONFIGURED` deve ser comentada com o caractere `#`, para que seja reconhecido que já foi efetuada a configuração e estará pronto para iniciar.

2.3.2.1 Dansguardian.conf

Para edição do `dansguardian.conf` utiliza-se um editor de texto comum, como mostra a figura 2.3.

```
# vi /etc/dansguardian/dansguardian.conf
```

Figura 2.3: Edição dansguardian.conf.

As principais opções a serem editadas podem ser vistas na Figura 2.4.

```
# Esta opção configura a língua em que as mensagens de bloqueio serão
#  mostradas no navegador para os usuários.
language = 'portuguese'

# Quantida de grupos possíveis.
filtergroups = 9

# Esta opção permite que seja repassado para os logs o logins
# dos usuários.
authplugin = '/etc/dansguardian/authplugins/proxy-basic.conf'

# Esta opção permite que os ips do usuários sejam repassados do squid
# para o DansGuardian.
forwardedfor = on
```

Figura 2.4: Alterações no arquivo de configuração do DansGuardian.

Maior detalhamento do arquivo `dansguardian.conf` encontra-se no apêndice A.2.

2.3.2.2 DansguardianfN.conf

Para cada grupo criado para o DansGuardian deve-se criar também um novo arquivo `dansguardianfN.conf`, em que N é o número do grupo, sendo que o grupo *default* recebe o numero 1. Neste arquivo temos a *tag* `naughtynesslimit`, que traduzida significa "índice de sem-vergonhice" (DansGuardian Organization, 2009).

Ao receber os arquivos das páginas, o DansGuardian verifica o conteúdo em busca de expressões e palavras "más", frequentemente encontradas em páginas indesejáveis, e também palavras "boas", normalmente encontradas em páginas de conteúdo adequado.

Cada palavra má soma certo número de pontos. Por exemplo, a expressão "menor" soma 8 pontos, enquanto a expressão "Menor de 18 anos" soma 90 pontos, palavras "boas" subtraem pontos, fazendo com que a página tenha possibilidade menor de ser bloqueada. A palavra "jurisprudência" subtrai 100 pontos, um bom exemplo para que uma página que contenha leis não seja bloqueada.

No `dansguardianfN.conf` existem também os caminhos das listas de onde serão feitas as consultas para bloqueios ou liberações de acesso aos grupos, as especificações das listas de bloqueios seguem relacionadas na Tabela 2.1 e as lista de exceções na Tabela 2.2.

Para o bom funcionamento, o administrador deve saber dosar a utilização dessas listas; o DansGuardian permite configuração fina, mas se mal configurado pode ter um funcionamento que não corresponda ao desejado (DansGuardian Organization, 2009).

Tabela 2.1: Arquivos de bloqueios utilizados pelo DansGuardian (SILVA; AUGUSTO, 2007).

Arquivos	Descrições das listas
`bannedphraselist`	Contém frases a ser bloqueadas
`bannedregexpurllist`	Bloqueio de expressões regulares
`bannedsitelist`	Sites a serem bloqueados
`bannedurllist`	URLs proibidas
`bannediplist`	IPs com acesso proibido
`bannedextensionlist`	Extensões de arquivos bloqueadas
`bannedmimetypelist`	Contém tipos de arquivos bloqueados
`headerregexplist`	Expressões regulares modificadas com *tags* bloqueadas, geralmente utilizadas em cookies
`bannedregexpheaderlist`	Expressões de cabeçalho a serem bloqueados
`weightedphraselist`	Frases bloqueadas
`picsfile`	Opções de filtros que contêm imagens
`contentregexplist`	Expressões regulares com *tags* bloqueadas
`urlregexplist`	Contém expressões regulares modificadas com tags bloqueadas, utilizadas em sites de busca

Tabela 2.2: Arquivos de exceções utilizados pelo DansGuardian (SILVA; AUGUSTO, 2007).

Arquivos	Descrições das listas
`exceptionfilesitelist`	Sites com downloads liberados
`exceptionfileurllist`	URLs com downloads liberados
`exceptionphraselist`	Frases liberadas
`exceptionsitelist`	Sites liberados
`exceptionurllist`	URLs, partes de sites liberados
`exceptionregexpurllist`	Permite liberar expressões regulares
`exceptionextensionlist`	Extensões de arquivos liberadas
`exceptionmimetypelist`	Contém tipos de arquivos liberados
`exceptioniplist`	IPs sem bloqueio algum
`greyurllist`	Habilita partes de sites, mas não desabilitas seus filtros
`greysitelist`	Permite habilitar sites, mas não desabilitas seus filtros

2.3.3 Principais Diretórios

Dentro do diretório principal do DansGuardian, o `/etc/dansguardian/`, há logo abaixo desse nível cinco diretórios: `authplugins`, `languages`, `lists`, `download`

managers e contentscanners. A Tabela 2.3 apresenta as descrições dos diretórios utilizados pelo DansGuardian.

Tabela 2.3: Diretórios de /etc/dansguadian/.

Diretório	Descrição
authplugins	Diretório de *plugins*, os mais utilizados são: IP que quando utilizado com X-Forwarded-For permite criar grupos de arranjos de ips e proxy-basic que repassam o nome dos usuários quanto autenticados;
contentscanner	Para utilização de antivírus, o padrão é o clamav;
downloadmanagers	Para utilização de gerenciadores de downloads;
languages	Tipos de idiomas das mensagem exibida no navegador;
lists	Arquivos de referência para configurações dos filtros.

No diretório lists, além dos arquivos de referência do dansguandianf1.conf, existem mais cinco diretórios, detalhados na Tabela 2.4.

Tabela 2.4: Diretórios de /etc/dansguadian/lists/.

Diretório	Descrição
authplugins	Arquivos utilizados pelos plugins, como listas de ips;
contentscanner	Arquivos utilizados para *scanner* com antivírus;
downloadmanagers	Arquivos utilizados por gerenciador de downloads;
phraselists	Listas de frases separadas por categorias e idiomas;
blacklist	Listas muito importantes para eficácia do filtro, podem-se utilizar listas atualizadas periodicamente em www.urlblacklist.org.

2.3.4 Criação e Configuração dos Grupos

Segundo a (DansGuardian Organization, 2009), no seu *changelog*, a partir da versão 2.9.2.0 foi excluída a lista exceptionuserlist, e na versão 2.9.3.0 foi adicionado *X-Forwarded-For*, que é o repasse dos IPs clientes ao Squid. Sem a exceptionuserlist para existir um grupo com acesso total, deve-se criar um grupo para os usuários desejados e no arquivo de configuração colocar a opção groupmode = 2.

Para criar um novo grupo, no diretório `/etc/dansguardian`, adiciona-se um novo diretório, que no caso chamará `grupo2`, e copiam-se os arquivos necessários de `/etc/dansguardian/lists/` para o diretório que foi criado, como pode ser observado na Figura 2.5.

```
# mkdir -p /etc/dansguardian/grupo2

# cd  /etc/dansguardian/lists/

# cp banned* contentregexplist exception* grey* headerregexplist
    pics weightedphraselist  urlregexplist ../grupo2
```

Figura 2.5: Criação de grupos no DansGuardian.

Comenta-se com o caractere `#`, dos arquivos copiados, as listas que não se desejam utilizar como restrições ao grupo. No diretório `/etc/dansguardian/` ainda deve ser feita uma cópia do arquivo `dansguardianf1.conf` com um novo nome, para ser utilizada pelo novo grupo, e editar o arquivo como mostra a Figura 2.6.

```
# cp dansguardianf1.conf dansguardianf2.conf

# vi dansguardianf2.conf
```

Figura 2.6: Criação e edição do dansguardianf2.conf.

No arquivo `dansguardianf2.conf`, observando a linha `bannedphraselist= /etc/dansguardian/lists/bannedphraselist`, verifica-se que aponta para o caminho `lists/bannedphraselist`; como houve a criação de um novo diretórios para este grupo, deve-se substituir `lists` por `grupo2`, no caso do Vi está substituição é efetuada com o comando `esc :%s/lists/grupo2/g`.

Prosseguindo a configuração, no parâmetro `naughtynesslimit` coloca-se um valor desejado para bloqueio por frase ponderada e também se deve editar o arquivo `dasnguardian.conf`, para indicar ao DansGuardian que um novo grupo foi criado. Não existe agora apenas o grupo *default*, e sim dois grupos, para completar essas alterações se troca o valor de `filtgroups` de 1 para 2.

2.3.4.1 Atribuições de Usuários aos Grupos

No diretório `/etc/dansguardian/lists` existe o arquivo `filtergroupslist`, para utilização do novo grupo, é necessário editar este arquivo fazendo atribuições aos usuários, como mostra a Figura 2.7. Após as alterações, é preciso reiniciar o DansGuardian com o comando `/etc/init.d /dansguardian restart` ou fazer apenas um *reload*, evitando a queda temporária do serviço, com o comando `dansguardian -r`.

```
# vim filtergrouplist

# Filter Groups List file for DansGuardian
#
# Format is <user>=filter<1-9> where 1-9 are the
groups
#
# This file is only of use if you have more than 1
filter group
#Atribua aqui o usuários do grupo 2, ex:
hendrigo=filter2
#Também pode ser atribuído a um ip:
192.168.0.100=filter2
```

Figura 2.7: Edição do filtergrouplist.

Criar grupos e atribuí-los aos usuários não é tarefa corriqueira que possa ser realizada por alguém que não tenha pleno conhecimento em DansGuardian e Linux, além do que há pouco material explicativo disponível na internet para essas tarefas. Dentre esses motivos surgiu a necessidade da criação de uma ferramenta que facilitasse e automatizasse tais funções em uma ferramenta *web*, onde o administrador necessitasse apenas de um navegador login e senha para executá-las. Dessas necessidades nasce o PhpDansAdmin, ferramenta descrita neste estudo.

2.4 Servidor Apache com Suporte a PHP

O servidor *web* utilizado na implementação deste trabalho, subdivide-se em duas grandes famílias: Apache 2.x e Apache 1.3, que apesar de muito antigo ainda é bastante utilizado. A versão 2 traz muitas vantagens, sobretudo do ponto de

vista de desempenho, além de oferecer novos módulos e mais opções de segurança (MORIMOTO, 2008).

A instalação do Apache 2 com suporte a PHP é bem simples: utilizando-se o gerenciador de pacotes `apt` do Debian executa-se o comando como na Figura 2.8.

```
# apt-get install apache2 apache2-utils php5
```

Figura 2.8: Instalação do Apache com suporte a PHP no Debian.

Efetuada a instalação, uma versão básica do Apache simplesmente exibe os arquivos `html, php` e executa scripts CGI. Para apresentar o funcionamento do Apache com o PHP, deve-se criar um arquivo `info.php` e salvá-lo em `/var/www` com o conteúdo do exemplo da Figura 2.9.

```
<?php
   phpinfo();
?>
```

Figura 2.9: Teste do Apache com suporte a PHP no Debian.

Por padrão o diretório raiz do servidor *web* é `/var/www` por esse motivo as páginas *web* do servidor "`http://ipdoseuservidor`*web*`/info.php`" é na verdade o arquivo "`/var/www/info.php`". O diretório raiz é definido por meio de uma opção dentro do arquivo principal de configuração (a opção `DocumentRoot`) e pode ser alterado como desejado.

2.4.1 Configurações do Apache

Umas das principais características do Apache é a modularidade estendida aos arquivos de configuração. Na versão 1.3, as configurações eram feitas no arquivo `httpd.conf`, que opcionalmente inclui referências a arquivos externos que permitem segmentar e organizar a configuração. Utilizando-se dessa possibilidade o Debian desenvolveu uma organização também utilizada nas distribuições variantes como, por exemplo, o Ubuntu (MORIMOTO, 2008).

A organização dos diretórios nas distribuições variadas do Debian, na primeira impressão parece ser mais complicada, mas após ser entendida sua estruturação, como demonstrado na Figura 2.10 se revela simples e lógica.

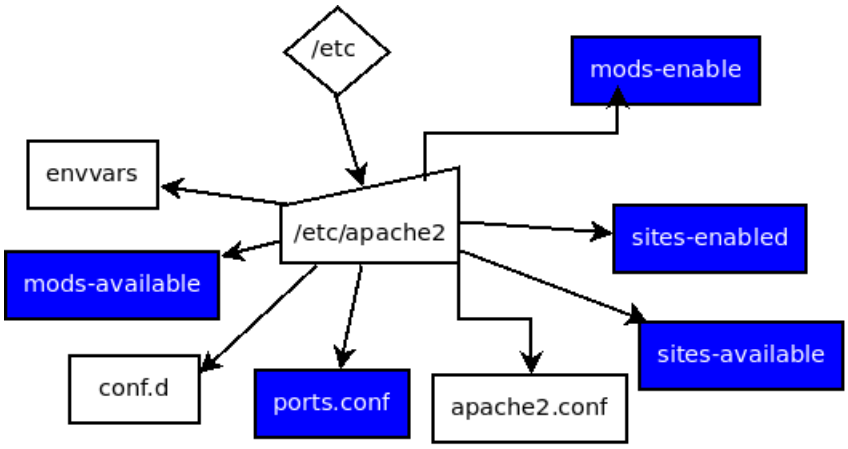

Figura 2.10: Estruturação dos arquivos do Apache (MORIMOTO, 2008).

As modificações que deverão ser efetuadas para implantar o PhpDansAdmin são basicamente: alteração do usuário *default* do Apache para um usuário com `bash` válido e a alteração no `DocumentRoot`.

A primeira ação é adicionar um usuário para ser utilizado pelo Apache como pode ser visto na Figura 2.11.

Com o usuário `suporte` criado, deve-se editar o arquivo `envvars` presente no diretório `/etc/apache2/` e alterar as variáveis de usuário e grupo do Apache, mantendo seu conteúdo como exibe a Figura 2.12.

Com esta configuração, o Apache é executado pelo usuário `suporte`, que após reinicialização poderá ser verificado, por exemplo, com o comando `ps` com as opções `-aux`.

No segundo passo, edita-se o arquivo `000-default` que se encontra no diretório `/etc/apache2/sites-enabled`, alterando a *tag* `DocumentRoot`, como mostrado na Figura 2.13.

Com a alteração no `DocumentRoot` os arquivos das páginas *web* estarão dentro do `home` do usuário `suporte`, o qual será o usuário do Apache, eliminando problemas com permissões referentes ao proprietário dos arquivos. Essa configuração no Apache é o modelo mais personalizado para o funcionamento do PhpDansAdmin.

```
:~# adduser suporte
Adicionando o usuário 'suporte' ...
Adicionando novo grupo 'suporte' (1002) ...
Adicionando novo usuário 'suporte' (1002) ao grupo 'suporte' ...
Criando diretório pessoal '/home/suporte' ...
Copiando arquivos de '/etc/skel' ...
Digite a nova senha UNIX:
Redigite a nova senha UNIX:
passwd: senha  atualizada com sucesso.
Modificando as informações de usuário para suporte
Informe o novo valor ou pressione ENTER para aceitar o padrão
        Nome Completo []: suporte Apache
        Número da Sala []: 0
        Fone de Trabalho []: 0
        Fone Doméstico []: 0
        Outro []: 0
Esta informação está correta?[s/n] s
```

Figura 2.11: Criação do usuário `suporte` no Debian.

```
# envvars - default environment variables for apache2ctl
# Since there is no sane way to get the parsed apache2
# config in scripts, some settings are defined via
# environment variables and then used in apache2ctl,
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=suporte
#export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=suporte
#export APACHE_RUN_GROUP=www-data
export APACHE_PID_FILE=/var/run/apache2.pid
```

Figura 2.12: Alteração do usuário do Apache.

2.5 OpenSSH

A ferramenta padrão para acesso remoto às máquinas equipadas com um sistema GNU/Linux é o OpenSSH, que permite a operação do terminal da máquina remota exatamente como se a operação fosse local. A principal diferença em relação a outras ferramentas de acesso remoto é a forte preocupação com a segurança e criptografia dos dados (SIQUEIRA, 2009).

```
<VirtualHost *:80>
        ServerAdmin \textit{web}master@localhost

        DocumentRoot /home/suporte/phpdansadmin
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
       ...
```

Figura 2.13: Alteração do DocumentRoot.

Apesar de ser um item fundamental na manutenção e operação segura de servidores remotos, o próprio OpenSSH não é totalmente livre de brechas de segurança, por isso é muito importante que o servidor e cliente estejam sempre atualizados para assegurar que as falhas encontradas foram corrigidas.

A instalação do OpenSSH é extremamente simples em um sistema GNU/Linux Debian e derivados. Para a instalação deve ser executado o comando `apt-get install ssh rsync`. Junto com o OpenSSH instala-se também o Rsync, uma ferramenta que será necessária na sincronização de arquivos e que utiliza o `ssh`.

O OpenSSH será utilizado pela ferramenta desenvolvida neste trabalho para executar comando remotos e será a ponte de sincronização dos arquivos alterados do servidor *proxy*. Para cumprir seu papel, deve-se configurar uma conexão `ssh` chaveada entre os servidores *web* e o *proxy*.

2.5.1 Autenticação por Chave entre Servidores

É a partir da chave criptográfica que o OpenSSH determina a confiabilidade e o método de criptografia usado numa conexão segura. As chaves criptográficas para o computador são geradas automaticamente pelo servidor `ssh`. Os arquivos para armazenar a chave privada e a chave pública variam de acordo com o tipo de criptografia utilizado, como mostra a Tabela 2.5.

Na primeira vez que o cliente `ssh` conecta-se a um servidor remoto, o usuário é questionado sobre aceitar a chave pública do computador remoto. Se for aceita, ela será armazenada em `/home/suporte/.ssh/know_hosts` e garantirá a confiabilidade de conexão entre os dois computadores. O conteúdo desse arquivo pode ser incluído no arquivo `/etc/ssh_know_hosts`, para que a chave passe a valer

Tabela 2.5: Chaves OpenSSH (SIQUEIRA, 2009).

Formato	Chave privada	Chave pública
RSA	/etc/ssh/ssh_host_rsa_key	/etc/ssh/ssh_host_rsa_key.pub
DSA	/etc/ssh/ssh_host_dsa_key	/etc/ssh/ssh_host_dsa_key.pub

para todos os usuários. Ainda assim será necessário que o usuário forneça senha ao conectar-se ao destino.

Dessa forma, se outro computador falsificar o nome ou IP da máquina remota, o cliente SSH informará o usuário que a identificação do servidor mudou e não estabelecerá a conexão. Nesse caso só será possível fazer o login via SSH se o usuário apagar a chave pública original do servidor armazenada anteriormente no arquivo `/home/suporte/.ssh/know_hosts`.

Além das chaves do próprio computador, cada usuário pode possuir a própria chave pública e privada, utilizada para garantir autenticidade. Dessa forma, é possível fazer com que o acesso via SSH seja feito automaticamente sem necessidade de fornecer senha em todo *login*. Antes de conseguir fazer o login sem senha, é preciso que o usuário crie a chave pública e a chave privada.

O par de chaves é gerado nesse ambiente no momento da instalação do PhpDansAdmin, e a chave será criada utilizando o parâmetro `rsa`, que suporta um tamanho em bits maior, por exemplo, 4096.

2.6 Exemplos de Ferramentas Administrativas em PHP

Existem várias ferramentas *web* desenvolvidas em linguagem PHP que são utilizadas para configuração de serviços, dentre elas se destacam phpLDAPadmin e phpMyAdmin.

2.6.1 PhpMyAdmin

O phpMyAdmin é uma ferramenta de software livre escrito em PHP destinado a lidar com a administração do MySQL na *World Wide Web*. Suporta uma ampla gama de operações com o MySQL, das quais mais utilizadas suportadas pela interface

do usuário são: gestão de bases de dados, tabelas, campos, relações, índices, usuários, permissões, além de poder executar diretamente qualquer declaração SQL. Fornecendo uma interface gráfica poderosa para gerenciar MySQL, phpMyAdmin é uma das interfaces mais populares de aplicações abertas.

Utilizado por milhões de pessoas, o MySQL é o banco de dados aberto mais popular entre os desenvolvedores. O MySQL adquiriu essa grande popularidade em virtude de sua natureza de código aberto, desempenho, confiabilidade, robustez e suporte para múltiplas plataformas. No entanto, a fama também tem sido ajudada pela existência do phpMyAdmin, ferramenta de administração padrão que facilita o gerenciamento de banco de dados (DELISLE, 2004).

Dentre os recursos do phpMyAdmin se destacam, de acordo com a (PHPMYADMIN.ORG, 2010):

- *web* intuitiva;
- navegação sobre bases de dados, tabelas, campos e índices;
- criar, copiar, renomear e alterar bancos de dados, tabelas, campos e índices;
- executar, editar e adicionar qualquer declaração SQL, até mesmo *queries* em lote;
- fazer gerenciamento de usuários do MySQL e seus privilégios;
- gerenciar *stored procedures* e *triggers*;
- fazer importação de dados de CSV e SQL;
- fazer exportação dados para vários formatos: CSV, SQL, XML, PDF, ISO / IEC 26300 - OpenDocument Texto e Planilha, Word, Excel, LaTeXe outros;
- permite criar gráficos PDF do *layout* do banco de dados.

O phpMyAdmin é traduzido em 57 línguas e suporta tanto línguas LTR[5] e línguas RTL[6]. Foi escolhida como a melhor aplicação PHP em várias premiações e a cada ano ganha o SourceForge.net Community Choice Awards como "Melhor ferramenta ou utilitário para *sysadmins*". O phpMyAdmin é um projeto de mais

[5]Determina a direção do fluxo do texto ou tabela definidos da esquerda para a direita, do Inglês: Left-To-Right.

[6]Determina a direção do fluxo do texto ou tabela definidos da direita para a esquerda, do Inglês: Right-To-Left.

de dez anos e tem uma base de código estável e flexível (PHPMYADMIN.ORG, 2010).

2.6.2 PhpLDAPadmin

O phpLDAPadmin (também conhecido como PLA) é um cliente baseado em *web* e fornece acesso fácil e multilíngue para um servidor OpenLDAP. Seu visual em árvore hierárquica e pesquisa avançada é intuitivo para navegação e administração da árvore de um diretório. Uma vez que é uma aplicação *web*, funciona em várias plataformas, fazendo com que o servidor seja facilmente controlado a partir de qualquer localização (PHPLDAPADMIN.ORG, 2010).

O phpLDAPadmin é um navegador com muitos recursos para o profissional LDAP, dentre os quais se destacam:

- LDAP *browser* em formato de árvore hierárquica;
- faz cópia de entradas LDAP (faz copia até mesmo entre diferentes servidores);
- oferece recursividade, cópia de árvores inteiras;
- apaga entradas LDAP;
- recursivamente, pode apagar árvores inteiras;
- exibe e edita atributos de imagem (como jpeg);
- faz pesquisas LDAP (simples e avançadas);
- faz exportação LDIF e DSML;
- faz importação LDIF;
- renomeia entradas LDAP;
- gerencia *hashes* de senha de usuários (SHA apoia, criptografia MD5, Blowfish e md5crypt);
- faz incremento do números UID automaticamente;
- é disponibilizado em dez idiomas.

A Figura 2.14, apresenta o visual do phpLDAPadmin, com exemplo de acesso a uma base LDAP.

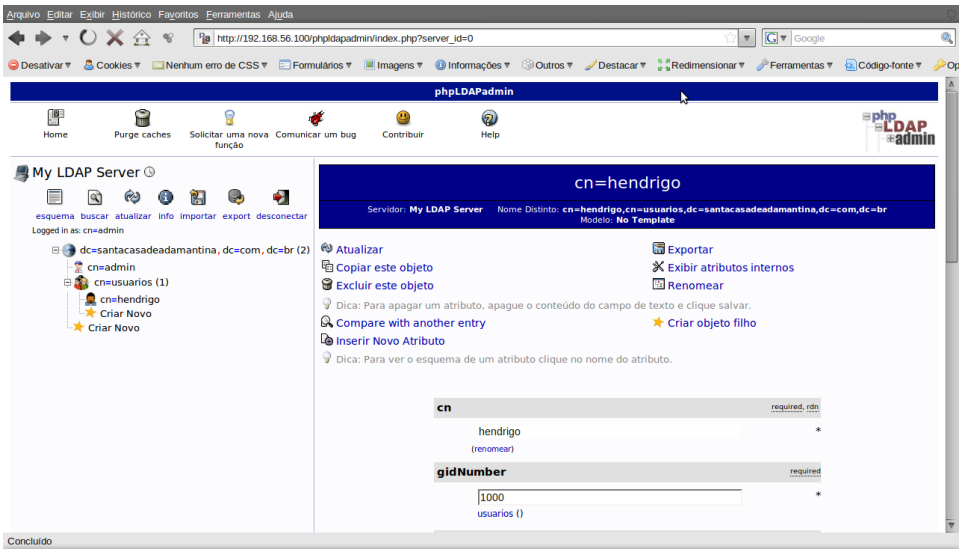

Figura 2.14: Exemplo de uso do phpLDAPadmin.

Capítulo 3

Desenvolvimento da Ferramenta

Este capítulo apresenta informações para o entendimento teórico e prático do desenvolvimento e funcionamento da ferramenta e também os caminhos e mecanismos criados para a execução completa de sua finalidade.

3.1 Idealização da Ferramenta

A execução das configurações nos perfis de acessos do DansGuardian é feita por intermédio de alterações em arquivos de formato texto, o que torna a tarefa árdua e mais propensa a erros. Devido a esse motivo e também por não existir atualmente uma ferramenta que viabilize tais tipos de configuração, surgiu a proposta do desenvolvimento de uma ferramenta que ofereça uma interface baseada em *web* que permita acessar de qualquer lugar no mundo, de forma personalizada, as configurações do DansGuardian.

Utilizando funções pré-definidas, é possível facilmente administrar os acessos de forma descentralizada. Instalada em um servidor Apache com suporte PHP, a administração do DanGuardian torna-se uma tarefa mais amigável, tais como outras ferramentas de administração *web* que podem estar também instaladas no mesmo servidor. Uma ilustração do escopo do funcionamento desses serviços pode ser vistos na Figura 3.1.

A Figura 3.1 representa a ação de um administrador efetuando operações como: criar usuários em uma base OpenLDAP utilizando o phpLDAPadmin, pesquisar em um banco de dados MySQL com o phpMyAdmin e fazer alterações de

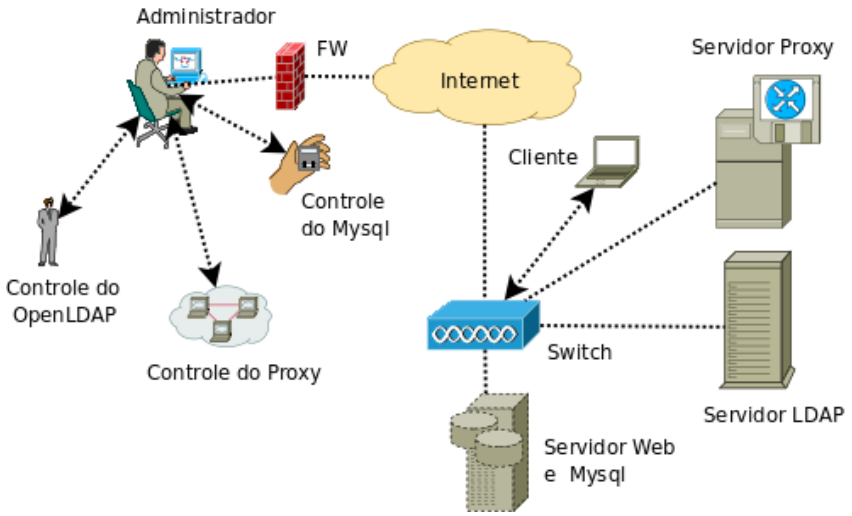

Figura 3.1: Representação de acesso a um servidor *web* com as ferramentas de administração.

perfis de acessos a usuários do *proxy*, por meio do PhpDansAdmin. Com esse escopo, é possível administrar vários serviços de uma rede, de qualquer lugar do mundo, utilizando uma conexão com a internet.

3.2 Detalhamento do Desenvolvimento

A ferramenta foi desenvolvida em linguagem PHP (Hypertext Preprocessor), utilizando a função `shell_exec()` para executar *scripts* em *Shell Bash*, o que a torna dependente de um servidor *web* GNU/Linux. A maioria dos comandos encontra-se nos códigos das páginas, com exceção de alguns que são comuns a vários tipos de execução e foram adicionados em scripts com extensão `.sh`. Os scripts encontram-se no diretório `bash` da ferramenta.

3.3 Scripts de uso Geral

Podendo ser utilizada tanto no mesmo servidor onde estiver instalado o DansGuardian como em um outro servidor *web* remoto, a ferramenta não faz alterações diretamente nos arquivos que estão sendo utilizados pelo DansGuardian, ela realiza um cópia fiel desses arquivos e após efetuadas as alterações é feita uma compara-

ção com os arquivos de origem e replicada no servidor somente as suas alterações, utilizando o aplicativo Rsync.

A Figura 3.2 mostra o arquivo `sinc-down.sh`, que utiliza o aplicativo Rsync para obter os arquivos do servidor DansGuardian em questão. Este script faz a cópia do diretório `/etc/dansguardian` para o diretório raiz do PhpDansAdmin, disponibilizando os arquivos para as alterações necessárias por meio da ferramenta *web*.

```
#!/bin/bash
# Sincroniza com o servidor, sentido DansGuardian - Web
# Gustavo Hendrigo Marcon
# Abril de 2010
rsync -ravzu  --partial --delete-excluded root@$1:/etc/dansguardian .
```

Figura 3.2: Script sinc-down.sh.

Na utilização do Rsync são incluídos os seguintes parâmetros: `r`, `a`, `v`, `z`, `u`, `partial` e `delete-excluded`, necessários para o funcionamento correto da aplicação. Os seus recursos são explicados na Tabela 3.1.

Tabela 3.1: Parâmetros utilizados pelo Rsync.

Parâmetros	Descrições
`r`	Cópia recursiva de diretórios, subdiretórios e seus conteúdos.
`a`	Copiar somente arquivos e diretórios.
`v`	Modo verbose.
`z`	Compactação de arquivos durante a transferência (e descompactar no destino).
`u`	Se o arquivo não foi atualizado, pula para o próximo, poupando tempo.
`partial`	Não copia arquivos quebrados.
`delete-excluded`	Também apaga arquivos e diretórios, e não só os sobrescreve.

No caminho contrário, a Figura 3.3 mostra o conteúdo do arquivo `sinc-up.sh`, que aplica as alterações no DansGuardian, sobrescrevendo os arquivos pelos arquivos alterados pela ferramenta. Finaliza corrigindo as permissões por meio do comando `chown` e faz o *reload* dos serviços do DansGuardian com a execução do comando remoto `dansguardian -r`, por intermédio do `ssh`.

```
#!/bin/bash
# Gustavo Hendrigo Marcon
# Abril de 2010
# Sincroniza com o servidor, sentido Web -> DansGuardian
rsync -ravzu --partial --delete-excluded $2/dansguardian root@$1:/etc/
# Corrige permissões
ssh root@$1 'chown -R dansguardian.dansguardian /etc/dansguardian'
# Reload no DasnGuardian
ssh root@$1 'dansguardian -r'
```

Figura 3.3: Script `sinc-up.sh`.

A ferramenta se dedica para que as sincronizações dos arquivos estejam exatas, em cada execução dos *scripts* de sincronia, é testada a comunicação entre os servidores web e *proxy*. A Figura 3.4 demonstra o código utilizado logo após o *login* na aplicação, ele executa a verificação de acesso no servidor *proxy* e caso haja alguma falha, retornará o erro ao usuário, ou caso contrário executará a sincronia.

```
<?php
// importa arquivo com variáveis estáticas
include ("conf/admin.php");

// Executa comando remoto para verificar acesso
$comando='ssh root@'.trim($proxy).' id -un';

// Resultado do comando
$servidor=shell_exec($comando);

// Condição sobre o resultado do comando
if ( (trim($servidor)) != "root") {
        echo "No conect to server ". $proxy;
        exit;
    }else{
    $pat=shell_exec("pwd");
    // Executa o script de sincronia
    $comando= trim($pat).'/bash/sinc-down.sh '.trim($proxy);
    }
?>
```

Figura 3.4: Código fonte para verificação de acesso ao servidor.

O arquivo `admin.php` exemplificado na Figura 3.4 contém a variável `$proxy`, que armazena o IP do servidor DansGuardian, informado no momento da execução da instalação da aplicação. O `admin.php` é incluído em boa parte dos fontes da aplicação, e através da variável citada são feitos os testes de acesso ao servidor *proxy*. No teste, é executado como superusuário o comando remoto `id -un`, armazenando o retorno na variável `$comando`, e sequencialmente é feita a seguinte condição: caso o conteúdo da variável é igual a `root`, significa que o comando remoto foi executado com sucesso e há a conexão com o servidor *proxy*.

3.4 Atualização da Lista Negra no DansGuardian

O DansGuardian, assim como as ACLs do Squid, também pode fazer uso de uma base de texto para consultas de URLs e domínios. O caminho padrão do diretório com as listas localiza-se em `/etc/dansguardian/lists/blacklists`. Neste diretório há vários subdiretórios classificados por categorias, que podem ser vistas no Apêndice B.1.

Cada subdiretório contém os arquivos de texto com os endereços que se desejam bloquear. A lista de arquivos com os conteúdos de bloqueio é conhecida mundialmente como lista negra ou *blacklist*[1], atualizadas constantemente pela comunidade que contribui com sua construção, o que torna seu uso indispensável.

É de boa prática ter uma *blacklist* sempre atualizada, um bom exemplo a ser citado seria o caso em que *proxies* externos são utilizados para burlar os sistemas de bloqueio da internet, estes tipos de domínios crescem em grande escala podendo surgir muitos novos diariamente.

Comumente as listas do projeto PhpDansAdmin são atualizadas semanalmente e o *download* do arquivo é feito a partir do site do projeto[2]. O arquivo nada mais é que uma cópia do endereço original da *blacklist*, porém há um teste a fazer na aplicação antes de ser disponibilizada.

Para a execução da atualização da lista foi criado o script `updateblacklist.sh`, que é invocado por meio da função `shell_exec()` do PHP, cujo detalhamento pode ser visto na Figura 3.5.

[1] Disponível em `https://www.urlblacklist.com`

[2] `https://sourceforge.net/projects/phpdansadmin/files/bigblacklist.tar.gz/download`

```
#!/bin/bash
# Update blacklist
# Gustavo Hendrigo Marcon
# Adamantina - SP, março de 2010
# Vars
PWD=`pwd`

wget -c https://sourceforge.net/projects/phpdansadmin/files/_
 _bigblacklist.tar.gz/download && tar -xvfz bigblacklist.tar.gz
if [ ! -e "$PWD/bigblacklist" ]
        then
        echo "Fail, don't updated!"
        exit 1
else
        mv blacklists ../dansguardian/lists -r
        date +%d/%m/%Y" "às" "%H:%M > data.txt
        echo "Updated with sucess!"
fi
```

Figura 3.5: Script `updateblacklist.sh`.

O script é bem simplificado ele faz o *download* do arquivo `bigblacklist.tar.gz` com a última versão disponível. Se for executado com sucesso, este arquivo é descompactado e movido para o local da lista antiga, sobrescrevendo-a.

3.5 Tratamento dos Grupos do DansGuardian

O DansGuardian permite a criação de até nove grupos, e em cada um pode ser personalizado um perfil de acesso. Na instalação é criado apenas um perfil, e a criação dos novos devem seguir o padrão do primeiro. Todos os usuários e IPs que não estiverem especificados na lista de designação de perfis, automaticamente pertencerão ao primeiro perfil, ou seja, o grupo *default* da aplicação.

3.5.1 Criação e Exclusão de Grupos

Na execução das ações de criar ou excluir um grupo, além das cópias e exclusão dos arquivos pertencentes a cada grupo, o parâmetro `filtergroups = 1` do `dansguardian.conf` deve ser incrementado ou decrementado. A função é execu-

tada pelos *scripts* `addgroup.sh` e `delgroup.sh` automaticamente. A Figura 3.6 demonstra o script `delgroup.sh`, que remove o grupo solicitado com o uso da ferramenta.

```
#!/bin/bash
# Fevereiro de 2010
# Gustavo Hendrigo Marcon

cd ../../
dir=$(pwd)
# Remove o diretório do grupo
rm -rf $dir/dansguardian/group$1
# Remove arquivo conf do grupo
rm -rf $dir/dansguardian/dansguardianf$1.conf
# Decrementa a quantidade de grupos
anterior=`expr $1 - 1`
# Subtrai a quantidade de grupos de filtergroups
sed -i "s/filtergroups = $1/filtergroups = $anterior/g"
            $dir/dansguardian/dansguardian.conf
```

Figura 3.6: Script `delgroup.sh`.

A Figura 3.7 exemplifica o script desenvolvido para criar os grupos executado através da função `shell_exec()` do PHP, passando como parâmetro o número do grupo a ser criado. O script basicamente faz as cópias de arquivos para os locais corretos e as configurações para o novo grupo.

Cada arquivo copiado, é uma lista que será manipulada pelo PhpDansAdmin, podendo ser especificado um perfil para cada grupo.

3.5.2 Manipulação dos Arquivos de Proibição do DansGuardian

Na implementação do PhpDansAdmin, foi incluído o tratamento dos arquivos que designam o bloqueio nos seguintes acessos: domínios, URLs, tipos e extensões de arquivo, expressões regulares, frases e IPs. Na lista de sites proibidos, por exemplo, existe a opção de bloquear por categorias, que são as categorias encontradas na *blacklist*, e também há opção para adicionadar sites individualmente.

Quando se referencia uma categoria, é incluído no arquivo `bannedsitelist` o caminho da lista através da *tag* `.Include< >` como, por exemplo:
`.Include</etc/dansguardian/lists/blacklists/proxy/domains>`

```
#!/bin/bash
# Fevereiro de 2010
# Gustavo Hendrigo Marcon

cd ../../
dir=$(pwd)
# Cria o diretório do grupo
mkdir -p $dir/dansguardian/group$1
sleep 1
# Copia os arquivos necessários
cp -p $dir/dansguardian/lists/banned* $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/contentregexplist $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/exception* $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/grey* $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/headerregexplist $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/pics $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/weightedphraselist $dir/dansguardian/group$1/
cp -p $dir/dansguardian/lists/urlregexplist $dir/dansguardian/group$1/
# Remove os arquivos inúteis
rm -rf $dir/dansguardian/group$1/bannediplist
rm -rf $dir/dansguardian/group$1/exceptioniplist
# Cria o arquivo de configuração
cp -p $dir/dansguardian/dansguardianf1.conf
          $dir/dansguardian/dansguardianf$1.conf
# Altera o arquivo de configuração
sed -i "s/lists/group$1/g" $dir/dansguardian/dansguardianf$1.conf
# Incrementa a quantidade de grupos
anterior=`expr $1 - 1`
# Troca a quantidade de grupos
sed -i "s/filtergroups = $anterior/filtergroups = $1/g"
           $dir/dansguardian/dansguardian.conf
```

Figura 3.7: Script `addgroup.sh`.

O arquivo `bannedsitelist` já contém a maioria dos caminhos das categorias, mas é recomendado adicionar todas as possíveis disponibilizadas na *blacklist*.

Os demais grupos criados são em primeiro momento uma cópia fiel do grupo *default* que posteriormente poderão ter configurados os perfis individualmente, por meio da aplicação. No exemplo de restrição por sites, a aplicação os separa em categorias e individualmente, porém estão em um mesmo arquivo o `bannedsitelist`, onde as tags `.Include< >` referenciam as categorias e os sites em si são relacionados no próprio arquivo.

```
# List other sites to block:
badboys.com

# To include additional files in this list use this example:
.Include</etc/dansguardian/lists/blacklists/adult/domains>
#.Include</etc/dansguardian/lists/blacklists/aggressive/domains>
#.Include</etc/dansguardian/lists/blacklists/artnudes/domains>
```

Figura 3.8: Trecho do arquivo `bannedsitelist`.

A Figura 3.8 apresenta um trecho do arquivo `bannedsitelist`, onde se podem observar as categorias já pré-adicionadas que podem ser habilitadas e desabilitadas na ferramenta, ou seja, as linhas com `.Include< >` são comentadas com o caractere `#` quando estão desabilitadas. O ato de incluir ou excluir sites, somente adicionam ou apagam novas linhas com os sites desejados.

3.5.3 Manipulação dos Arquivos de Exceções do DansGuardian

Quando são criados os grupos, geralmente são configuradas as categorias a que estes não terão acessos, mas há casos em que um site específico de uma categoria não deverá ser bloqueado. Nestes casos, eles podem ser adicionados em listas específicas para que não sejam bloqueados, nada mais justo serem denominadas listas de exceções.

O PhpDansAdmin trata as seguintes listas de exceções do DansGuardian: domínios, URLs, downloads de sites, downloads de URLs, extensões de arquivos, tipos de arquivo e IPs, que podem ser configuradas individualmente para cada grupo.

3.5.4 Associação de Usuários aos Grupos

Após a criação dos grupos e a personalização dos perfis de acesso, devem-se atribuir os grupos aos usuários. Os usuários em questão podem ser o *login* de autenticação ou até mesmo o IP do cliente. As referências dos grupos são feitas no arquivo `filtergroupslist` do DansGuardian, e o tratamento do arquivo é efetuado através das funções do PHP.

A Figura 3.9 referencia um trecho do código fonte utilizado para mostrar o conteúdo da lista. A lógica é simples: verifica-se a linha não tem o caractere inicial # , ou se é uma linha em branco, caso contrário é utilizada pelo DansGuardian e deve estar no formato `usuário = filterN`, em que N é o número do grupo especificado limitado de 1 a 9.

```
for($i=0; $i<sizeof($lines); $i++) {
   // Verifica se é um comentário ou linha em branco para pular a linha
   if(($lines[$i][0] == "#")||($lines[$i][1] == "")) {
       file_put_contents($arquivo, file_get_contents($arquivo) . $lines[$i]);
       continue;
   }
   // Verifica se é a linha à apagar
   if($i != $del) {
       // Salva a linha no arquivo
       file_put_contents($arquivo, file_get_contents($arquivo) . $lines[$i]);
   }
}
```

Figura 3.9: Tratamento do arquivo `filtergroupslist`.

A Figura 3.10 mostra o arquivo `filtergroupslist` editado pela ferramenta, observa-se que é preservada as linha comentadas e adicionado no final do arquivo os novos usuários com os respectivos grupos (filtros) associados.

```
# Filter Groups List file for DansGuardian
# Format is <user>=filter<1-9> where 1-9 are the groups
# Eg:
# This file is only of use if you have more than 1 filter group
#
hendrigo=filter2
gustavo=filter3
```

Figura 3.10: Arquivo filtergroupslist.

3.6 Edição dos Arquivos dansguardian.conf e dansguardianfN.conf

Os processos de tratamento dos arquivos `dansguardian.conf` e `dansguardianfN.conf` são basicamente iguais, a diferença entre eles é que o primeiro edita as

configurações do DansGuardian propriamente dita, e o segundo trata as configurações dos grupos, em que N, do nome do arquivo, é substituído pelo valor do grupo, com os valores de 1 a 9. Quando carregada a página de edição dos arquivos de configurações, os valores dos parâmetros contidos nos arquivos são carregados em variáveis com o auxilio da função `shell_exec()` do PHP. A Figura 3.11 apresenta o trecho onde são atribuídos os valores às variáveis no carregamento da página.

```
<?php
   // log level
   $comando='grep loglevel dansguardian.conf|grep =|cut -d = -f 2';
   $loglevel=shell_exec($comando);
   // language
   $comando='grep -w language dansguardian.conf|grep =|cut -d "\'" -f 2';
   $language=shell_exec($comando);
...
```

Figura 3.11: Código fonte, armazenando parâmetros do `dansguardian.conf`.

Com o resultado dos parâmetros nas variáveis, já é possível montar o formulário *web* para a edição do arquivo. Efetuadas as alterações pela ferramenta, é preciso gravá-las no arquivo novamente. A Figura 3.12 apresenta o código fonte que executa as trocas dos parâmetros com o auxilio da função `shell_exec()`, que executa o comando `sed` alterando os valores que são repassados pela página no arquivo `dansguardian.conf`.

```
<?php
   //var language
   $comando='grep -w language dansguardian.conf|grep =|
                            cut -d "\'" -f 2';
   $language_old=shell_exec($comando);
   $language_new=trim($_POST['language']);

   //Troca language
   $comando="sed -i 's/" . trim($language_old) . "/" . trim($language_new).
                           "/g' dansguardian.conf";
   shell_exec($comando);

   //var loglevel
   $comando='grep loglevel dansguardian.conf|grep =|cut -d = -f 2';
   $loglevel_old=shell_exec($comando);
   $loglevel_new=trim($_POST['loglevel']);

   //Troca loglevel
   $comando="sed -i 's/loglevel = " . trim($loglevel_old) . "/loglevel = " .
                           trim($loglevel_new). "/g' dansguardian.conf";
   shell_exec($comando);
...
```

Figura 3.12: Código fonte, trocando parâmetros do `dansguardian.conf`.

Capítulo 4

Instalação da aplicação

Este capítulo aborda os passos que devem ser executados na instalação do Php-DansAdmin e suas premissas.

4.1 Premissas

Para o funcionamento da ferramenta desenvolvida neste estudo, é necessário que estejam instalados e configurados os seguintes serviços:

- sistema operacional Linux, neste estudo foi utilizada a distribuição GNU Debian 5.0, versão 32 bits;
- servidor Apache com suporte a PHP 5, com seu serviço iniciado com um usuário com *bash* válido, e as permissões de execução e leitura nos diretório das páginas *web*, atribuídas ao usuário do Apache;
- servidor *proxy* que utilize o DansGuardian, instalado e primariamente configurado;
- aplicativos OpenSSH e Rsync instalados em ambos os servidores, no caso de os servidores *web* e *proxy* estarem em *hardwares* distintos.

Umas vez que os aplicativos requeridos estejam instalados e configurados, podem-se iniciar os passos de instalação.

4.2 Instalação

A instalação do PhpDansAdmin é bem simplificada. No primeiro passo é necessário descompactar o arquivo `phpdansadmin.tar.gz`[1], como mostra a Figura 4.1 (atentar ao diretório onde foi especificado no *DocumentRoot* do Apache), acessar a página `http://ipdoservidor/phpdansadmin/install/` e iniciar o processo de instalação.

```
suporte@ARL-seg:~ pwd
/home/suporte

suporte@ARL-seg:~ ls
phpdansadmin.tar.gz

suporte@ARL-seg:~ tar -xzvf  phpdansadmin.tar.gz phpdansadmin/
phpdansadmin/menu/
phpdansadmin/menu/index.html
phpdansadmin/menu/jdMenu-demo.html
phpdansadmin/menu/sair.php
phpdansadmin/menu/Conteudo.html
phpdansadmin/menu/iframes.html
...

suporte@ARL-seg:~  ls
phpdansadmin  phpdansadmin.tar.gz
suporte@ARL-seg:~  rm -rf phpdansadmin.tar.gz
```

Figura 4.1: Descompactação do arquivo `phpdansadmin.tar.gz`.

A Figura 4.2 exibe a tela inicial da instalação do PhpDansAdmin, na primeira etapa deve-se clicar no botão `Next` e acessar a próxima página de instalação.

[1] Disponível em `https://sourceforge.net/projects/phpdansadmin/`

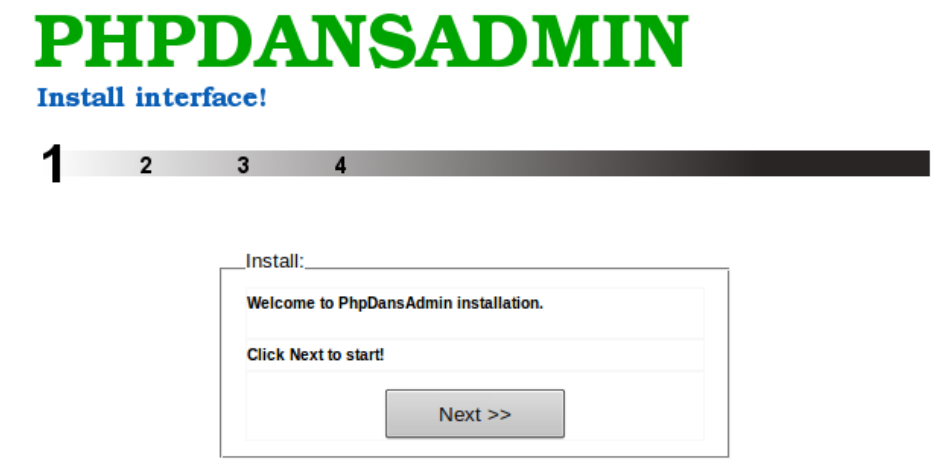

Figura 4.2: Instalação do PhpDansAdmin, passo 1.

No segundo passo da instalação é necessário informar um *login*, o qual será o administrador do ambiente. Não é obrigatório utilizar o *login* admin ou sinônimos, é possível criar qualquer denominação.

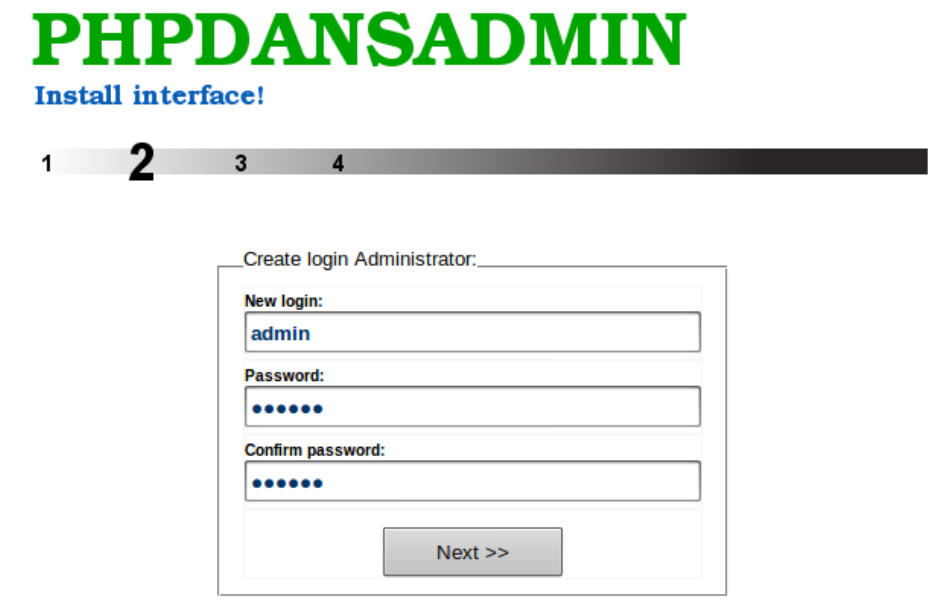

Figura 4.3: Instalação do PhpDansAdmin, passo 2.

O terceiro passo irá configurar e testar a conexão com o servidor onde estará o DansGuardian. Primeiramente, enquanto não há a conexão `ssh` chaveada, será apresentada a tela como está representada na Figura 4.4, que indica que é necessário executar o script `install-ssh.sh`, que se encontra no diretório /phpdansadmin/install.

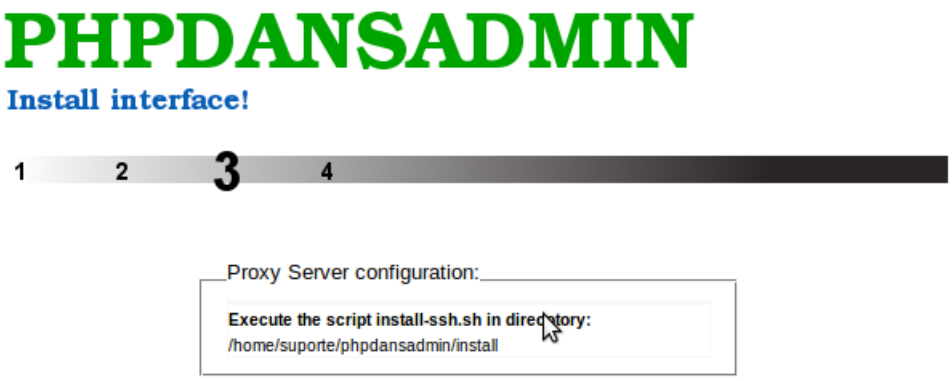

Figura 4.4: Instalação do PhpDansAdmin, passo 3.1.

A Figura 4.5 demonstra a execução do script `install-ssh.sh`, que irá criar a conexão chaveada do usuário do Apache com o servidor *proxy*.

Ao executar o script deve-se teclar *enter* nas primeiras perguntas até ser solicitada a senha de *root* do servidor onde se encontra o DansGuardian, e em um segundo passo redigitar a senha. O script faz a verificação se está sendo executado com o mesmo usuário do Apache e também se o IP passado como parâmetro é válido.

O IP é validado por intermédio de uma expressão regular. As expressões regulares, também conhecidas como *regexp*, são um conjunto de caracteres alfanuméricos e/ou metacaracteres, que combinados entre si de acordo com algumas regras são capazes de extrair informações dentro de textos (CAMARGO, 2005).

```
suporte@web:~/phpdansadmin/install$ ls
admin.php    execserver.php  flag       install-ssh.sh  server.php
execadm.php  fim.php         index.php  jainstal.php
suporte@web:~/phpdansadmin/install$ ./install-ssh.sh 192.168.56.2
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
Loading...
Generating public/private rsa key pair.
Enter file in which to save the key (/home/suporte/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/suporte/.ssh/id_rsa.
Your public key has been saved in /home/suporte/.ssh/id_rsa.pub.
The key fingerprint is:
35:8c:68:a3:13:ff:c2:ac:01:cf:7e:79:9d:ce:41:74 suporte@web
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|       . o       |
|    . + . = E    |
|     = . o o     |
|  . o . S .      |
|   + + . .       |
|    + +.....     |
|   . oo...o.     |
|    o. . .o      |
+-----------------+
TYPE PASSWORD OF ROOT OF THE 192.168.56.2
The authenticity of host '192.168.56.2 (192.168.56.2)' can't be established.
RSA key fingerprint is 40:85:d4:a0:cd:f0:98:f9:7b:27:f6:38:d3:db:4c:b7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.2' (RSA) to the list of known hosts.
root@192.168.56.2's password:
RETYPE PASSWORD
root@192.168.56.2's password:
suporte@web:~/phpdansadmin/install$
```

Figura 4.5: Instalação do PhpDansAdmin, passo 3.2.

Após a execução do script, se a conexão for bem-sucedida, será apresentado automaticamente a tela como mostra a Figura 4.6, caso não seja apresentada, será necessário executar novamente o script `install-ssh.sh`.

Finalizada a instalação, serão efetuados os testes para verificar se o PhpDansAdmin pode executar as tarefas. Caso esteja corretamente configurado exibirá uma tela, como demonstrado na Figura 4.7.

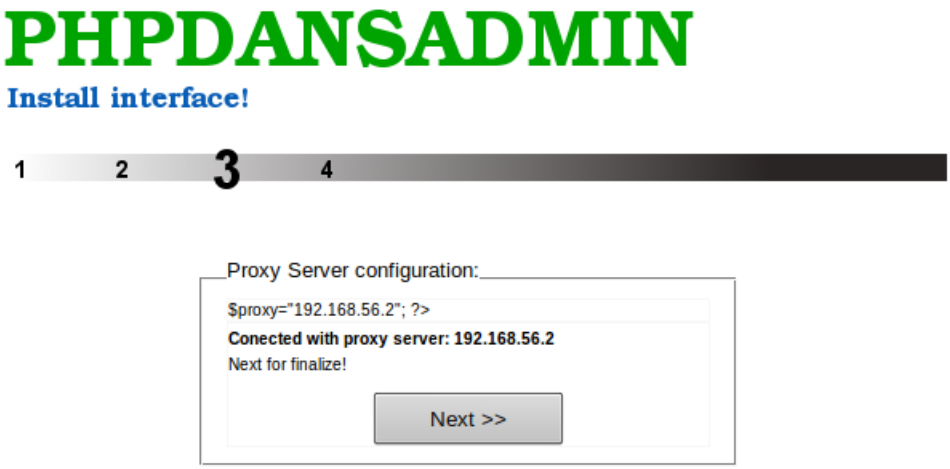

Figura 4.6: Instalação do PhpDansAdmin, passo 3.3.

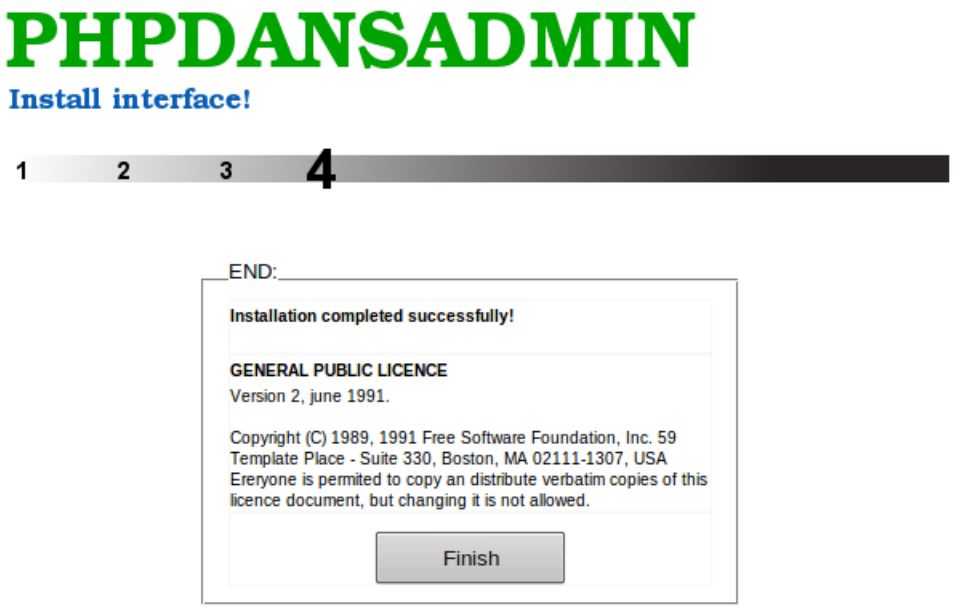

Figura 4.7: Instalação do PhpDansAdmin, finalização.

Ao clicar no botão `Finish`, a aplicação estará pronta para utilização, e irá trazer a página para *login*.

4.3 Segurança em uma Nova Instalação

Por questões de maior segurança, ao acessar novamente a URL: `http://ipdoservidor/phpdansadmin/install`, será apresentada a tela representada na Figura 4.8, isso evita a alteração de senha de administrador por qualquer pessoa mal-intencionada.

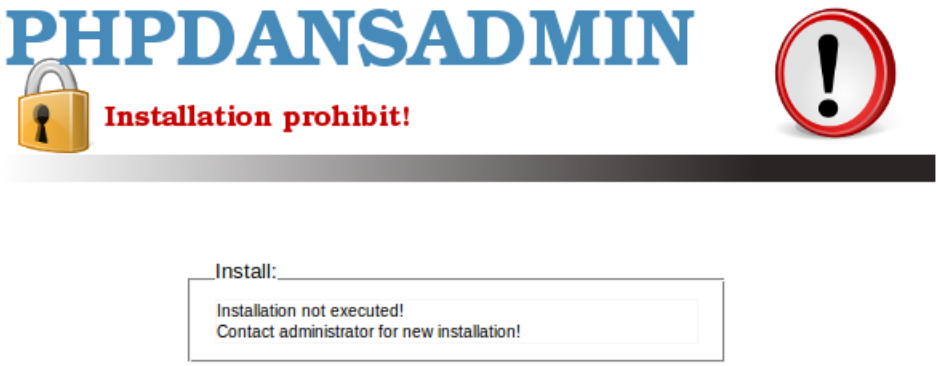

Figura 4.8: Tela de aviso quando a instalação já foi executada.

Para executar nova instalação, em que será possível a criação de um novo *login* e senha para o administrador, deve-se acessar a aplicação e executar o item `New installation` do menu `Help`. Após esses passos será possível executar a instalação normalmente.

O capítulo a seguir apresenta um caso de uso, demonstrando na prática como a aplicação minimiza o nível de complexidade da administração dos perfis de acessos do DansGuardian, com uma rotina bem comum utilizada por um administrador.

Capítulo 5

Testes e Resultados

Neste capítulo, será apresentado um teste na aplicação com um caso de uso, onde será criado um novo grupo, alterado o seu arquivo de configuração e as categorias de sites que devem ser bloqueados.

No teste será mostrado a associação de um usuário a um grupo, o resultado de bloqueio ao tentar acessar um site de uma das categorias que foram proibidas ao grupo e a alteração da linguagem utilizada pelo DansGuardian.

5.1 Ambiente de Teste

Para o ambiente de testes, foi utilizado a distribuição Debian 5.0, Apache 2 com PHP5, Squid 2.7 e DansGuardian 2.9, todos devidamente instalados e configurados. Nos testes realizados, a ferramenta se mostrou bastante funcional, permitindo acesso fácil às opções e oferecendo interfaces intuitivas para a interação do usuário.

5.2 Login no PhpDansAdmin

No primeiro passo, é necessário acessar a página onde se encontra a aplicação e digitar o *login* do administrador e senha, criados na execução da instalação.

A Figura 5.1 demonstra um exemplo de acesso à aplicação.

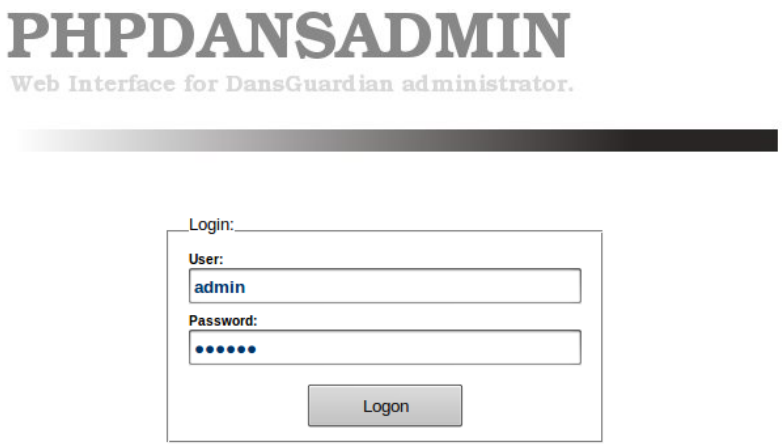

Figura 5.1: Tela inicial, login do administrador.

5.3 Página Inicial Após o Login

Após o *login* será também verificado a conexão com o servidor *proxy*, se todos os dados e execuções forem efetuados com sucesso, será exibida a página como demonstrado na Figura 5.2.

Figura 5.2: Página inicial do PhpDansAdmin.

5.4 Adicionar um Grupo

O DansGuardian, na instalação, já cria o que pode ser denominado de Grupo 1, ou grupo *default*. Em uma primeira tarefa pode-se criar um grupo seguinte, o Grupo 2. Para a execução é preciso acessar os menu `Add Groups` e clicar no *link* habilitado (Group 2), como demonstra a Figura 5.3.

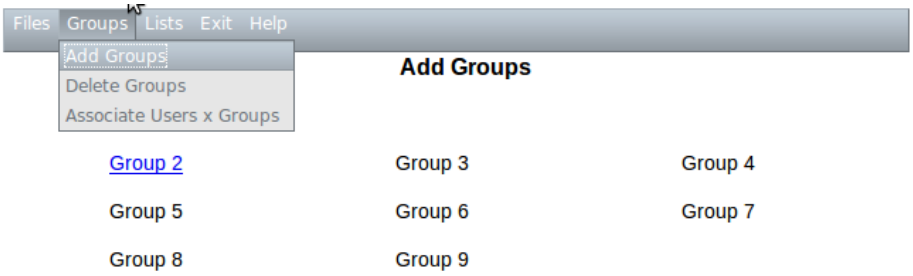

Figura 5.3: Criação de um novo grupo.

A criação dos grupos assim como a exclusão devem ser executadas sequencialmente, devido ao acréscimo ou decréscimo na variável `filtergroups` do arquivo `dansguardian.conf`. Esse mecanismo é feito automaticamente pela ferramenta, ela só deixará ser criado o Grupo 3, por exemplo, se já existir o Grupo 2, e assim por diante.

5.5 Customização do Arquivo de Configuração do Grupo

Na criação de um grupo, também é criado o arquivo de configuração específico, que pode ser alterado de acordo com as necessidades dos perfis de acessos do grupo. Deve-se acessar o menu `DansGuardian.confs` para fazer as alterações necessárias.

A Figura 5.4 exibe os menus de acesso dos arquivos de configurações e a Figura 5.5 demonstra uma alteração efetuada no parâmetro `Naughtyness Limit`, que foi alterado do padrão 50 para 160.

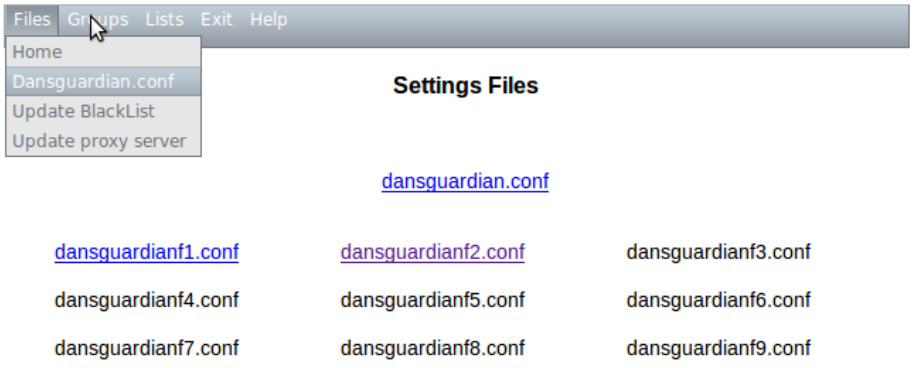

Figura 5.4: Menu dos arquivos de configurações.

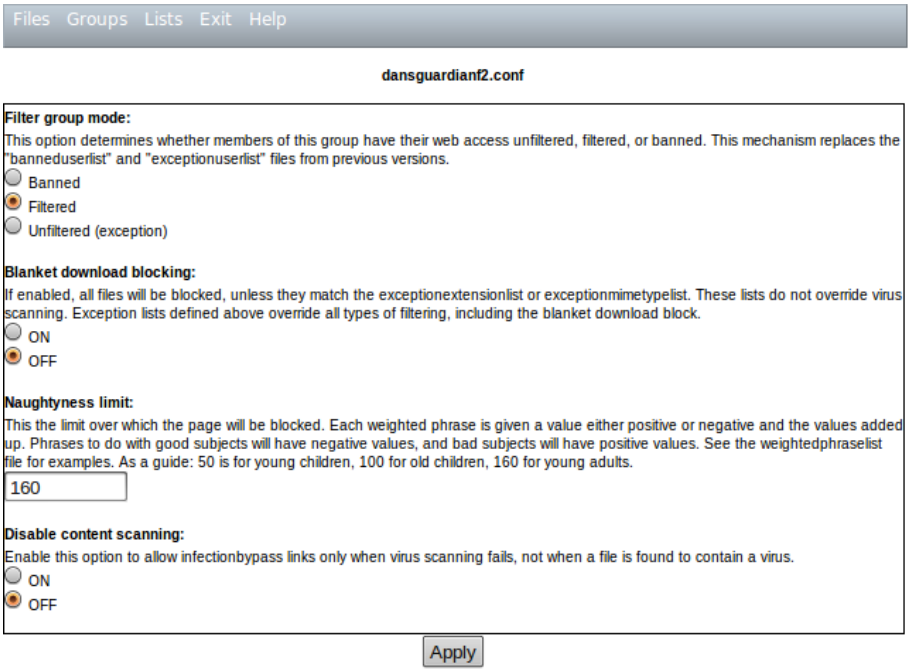

Figura 5.5: Alteração no `dansguardianf2.conf`.

5.6 Associação de Usuários a Grupos

Por definição, se um usuário não está associado a nenhum grupo, ele pertencerá ao *grupo default* da aplicação. No exemplo aqui apresentado, foi criado num novo grupo (Group 2). Para que este grupo seja utilizado é necessário que seja associado

um usuário a ele. Essa função deve ser executada acessando o menu `Associate Users X Groups` como demonstra a Figura 5.6.

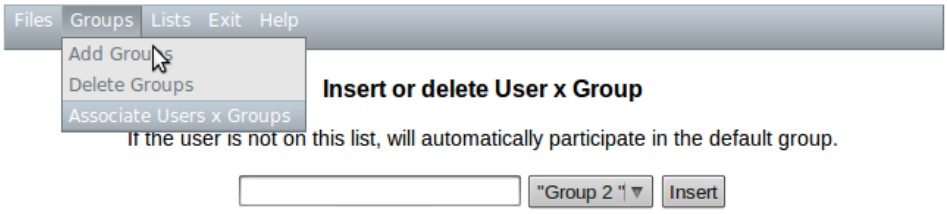

Figura 5.6: Acesso a página de associação de usuários.

Como apresenta a Figura 5.7, para fazer a associação deve-se digitar um *login* de acesso utilizado no servidor *proxy*, selecionar um dos grupos disponíveis, teclar `Enter` ou clicar em `Insert`.

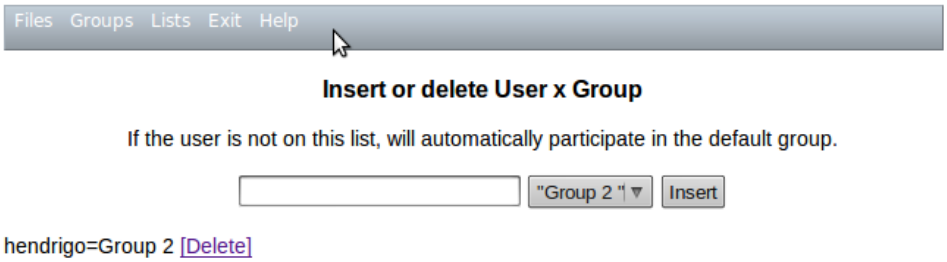

Figura 5.7: Associação de usuário ao grupo.

5.7 Edição do Perfil de Acesso de um Grupo

Para a edição do perfil de um grupo, devem-se acessar os menus de bloqueios ou exceções e configurar os acessos como desejado. A Figura 5.8 demonstra um acesso ao menu de categorias de sites bloqueados, que na prática habilita ou desabilita umas das `blacklists`.

A Figura 5.9 exibe a habilitação da categoria de *chat*, ou seja, os sites referentes a *chat* contidos na `blacklist` serão bloqueados para este grupo de usuários, neste caso representado pelo usuário `hendrigo`.

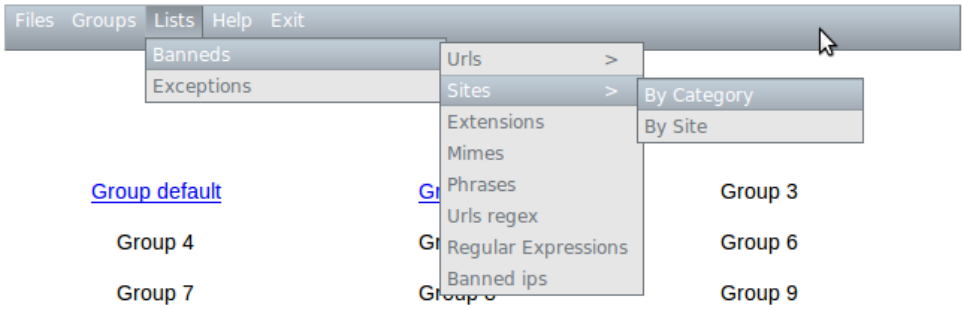

Figura 5.8: Menu de acesso a categorias de sites bloqueados.

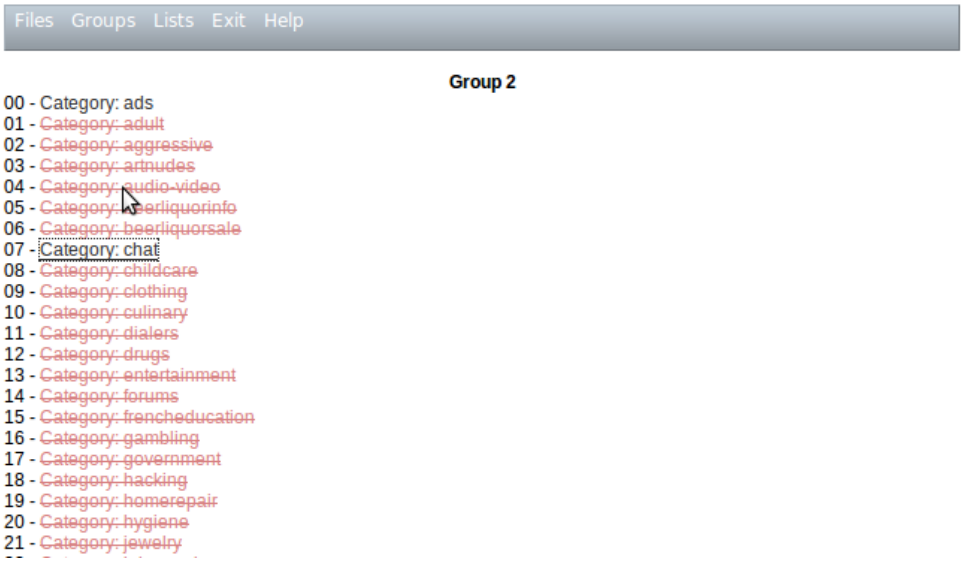

Figura 5.9: Habilitação da categoria *chat*.

O arquivo referente a esta alteração, o `bannedsitelist` do `Group 2`, é demonstrado na Figura 5.10. Nota-se que as listas habilitadas na aplicação são idênticas ao arquivo referido.

Após as customizações dos perfis, é necessário aplicá-las no servidor *proxy*, para isso se deve clicar no menu `File - Update proxy server` ou em `Exit - Exit and update proxy server`. Na primeira opção serão aplicadas as altera-

```
# The squidGuard advert domain/URL lists are now included by default.
# To work with advanced ad blocking & the logadblocks option, advert
# phrase/site/URL lists should have the string "ADs" in their listcategory.
#Remove the # from the following and edit as needed to use a sto
.Include</etc/dansguardian/lists/blacklists/ads/domains>
#.Include</etc/dansguardian/lists/blacklists/adult/domains>
#.Include</etc/dansguardian/lists/blacklists/aggressive/domains>
#.Include</etc/dansguardian/lists/blacklists/artnudes/domains>
#.Include</etc/dansguardian/lists/blacklists/audio-video/domains>
#.Include</etc/dansguardian/lists/blacklists/beerliquorinfo/domains>
#.Include</etc/dansguardian/lists/blacklists/beerliquorsale/domains>
.Include</etc/dansguardian/lists/blacklists/chat/domains>
#.Include</etc/dansguardian/lists/blacklists/childcare/domains>
#.Include</etc/dansguardian/lists/blacklists/clothing/domains>
#.Include</etc/dansguardian/lists/blacklists/culinary/domains>
#.Include</etc/dansguardian/lists/blacklists/dialers/domains>
#.Include</etc/dansguardian/lists/blacklists/drugs/domains>
#.Include</etc/dansguardian/lists/blacklists/entertainment/domains>
#.Include</etc/dansguardian/lists/blacklists/forums/domains>
#.Include</etc/dansguardian/lists/blacklists/frencheducation/domains>
#.Include</etc/dansguardian/lists/blacklists/gambling/domains>
#.Include</etc/dansguardian/lists/blacklists/government/domains>
#.Include</etc/dansguardian/lists/blacklists/hacking/domains>
                                                          65,1          71%
```

Figura 5.10: Arquivo `bannedsitelist` referente ao grupo 2.

ções e o PhpDansAdmin será mantido para demais customizações, já na segunda o administrador sairá da ferramenta aplicando as alterações no servidor *proxy*.

5.8 Teste das Alterações no DansGuardian

Um teste das alterações efetuadas pode ser executado através da tentativa de um acesso como o usuário específico. A Figura 5.11 exibe o *login* de um usuário pertencente ao Grupo 2, na tentativa de acesso ao site `http://chat.terra.com.br`.

As alterações nos arquivos do DansGuardian bem como o *reload* foram efetuados com sucesso, isso pode ser notado na Figura 5.12, que exibe o bloqueio na tentativa de acesso ao site de *chat*.

Ajustes mais finos podem ser executados dando a possibilidade de criação de até nove perfis, que é a limitação do DansGuardian na versão utilizada. Parece ser uma quantidade pequena mas em se tratando de perfis de acesso, em que os tipos de site bloqueados geralmente se repetem, e sendo que a cada perfil pode haver inúmeros usuários, isso se torna um leque grande de opções.

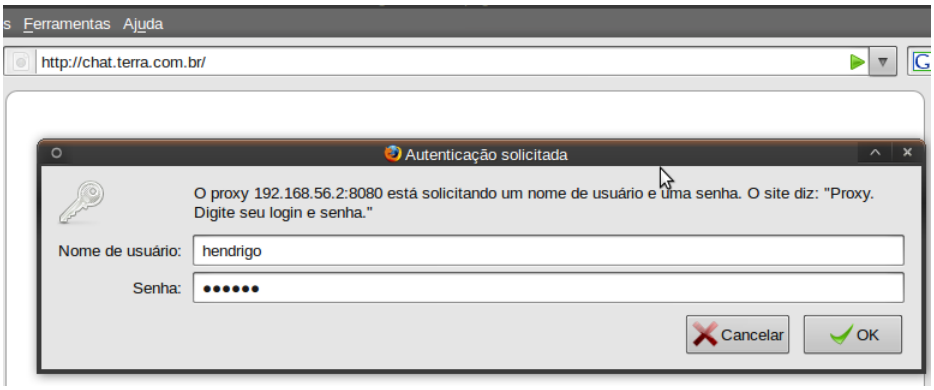

Figura 5.11: Login de usuário no Proxy.

Figura 5.12: Tela de bloqueio do DansGuardian.

5.9 Alterações nas Configurações do DansGuardian

Após a instalação e configuração do DansGuardian, não é muito comum ocorrer várias alterações no arquivo `dansguardian.conf`, porém a ferramenta desenvolvida suporta algumas das alterações mais comuns que por ventura possam ser solicitadas.

A Figura 5.13 demonstra a alteração do idioma que o DansGuardian deverá trabalhar: neste caso houve alteração da língua portuguesa para a espanhola. O resultado dessa ação pode ser visto na Figura 5.14, que é a exibição das mensagens na nova língua escolhida.

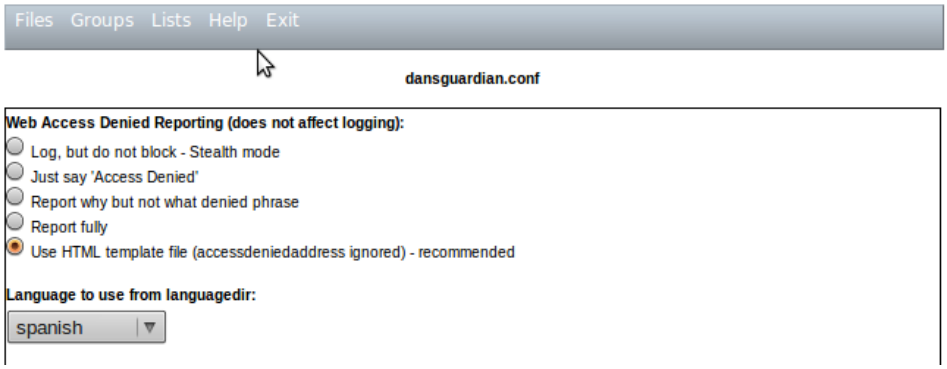

Figura 5.13: Alteração do idioma no DansGuardian.

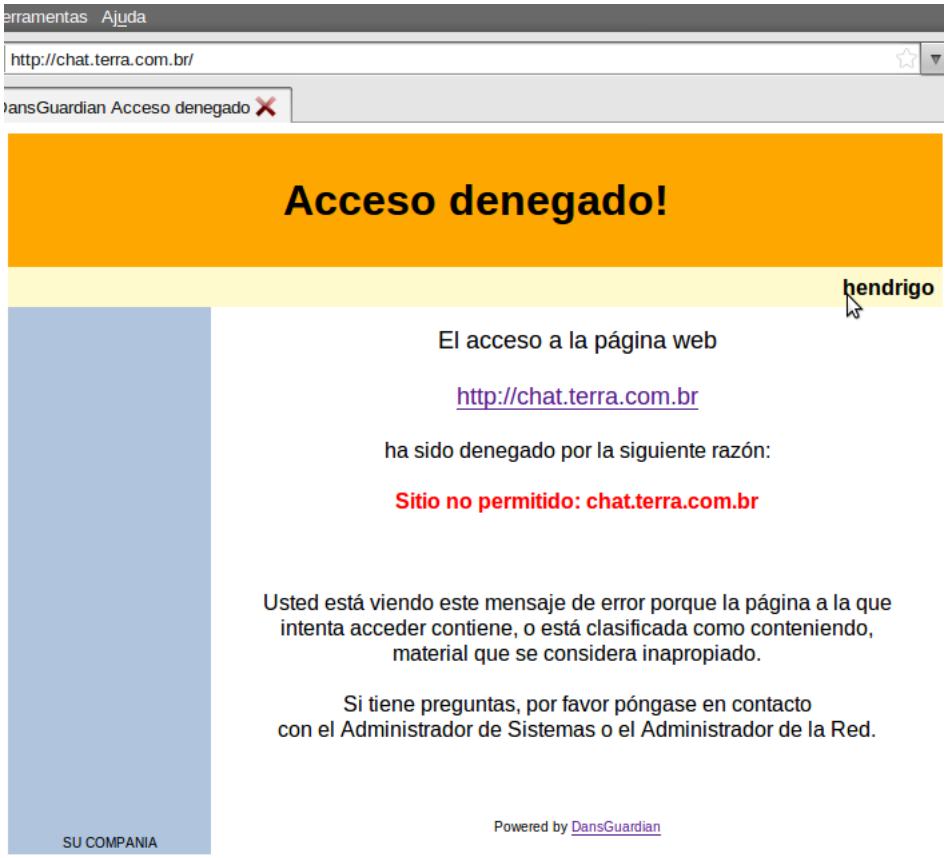

Figura 5.14: Resultado da alteração do idioma no DansGuardian.

Capítulo 6

Conclusão

Segundo (UCHÔA & UCHÔA, 2007), o desenvolvimento de aplicações é um tipo de pesquisa bastante comum que apresenta soluções para determinado problema organizacional já diagnosticado.

Tornar as tarefas de administração mais rápidas e simples de serem realizadas e com menos propensão a erros viabiliza maior versatilidade na criação de regras e eficácia na administração dos acessos à rede.

O PhpDansAdmin pretende atender de forma satisfatória suas intenções, simplificando a administração do DansGuardian e criando a possibilidade de um administrador, com menor conhecimento, gerenciar regras de filtragem.

O grau de abstração atingido permite que o administrador realize de forma prática operações como criar e alterar arquivos de configuração da forma desejada, atualizar *blacklists*, criar e excluir grupos, associar usuários a grupos, administrar os perfis dos grupos por meio das listas de URLs, domínios, extensões, tipos de arquivo, expressões regulares, frases e combinações de palavras proibidas e liberadas.

Em um ambiente, por exemplo, em que há alta rotatividade de funcionários, onde existe a necessidade de relacionar constantemente os usuários aos grupos existentes, o PhpDansAdmin pode ser de grande utilidade, facilitando em muito a administração da rede e economizar tempo e diminuir erros.

A ferramenta apresentada neste trabalho é primeira versão. Eventuais melhorias poderão ser realizadas pelo autor ou pela comunidade *Open Source*. Seu

código fonte está disponibilizado e licenciado sob a `General Public Licence - (GPL)`, e pode ser encontrado em: https://sourceforge.net/projects/phpdansadmin/.

6.1 Trabalhos Futuros

Sugere-se como futuros desenvolvimentos, a implantação dos seguintes itens:

- utilizar uma base LDAP, para seleção de usuários;
- melhorar o visual da interface;
- aplicar criptografia na senha de administrador;
- fazer adaptação a novas versões do DansGuardian;
- fazer instalação totalmente via *web*, sem a necessidade de execução do script `install-ssh.sh` pelo terminal;
- desenvolvimento de perfis de acessos administrativos;
- testes de vulnerabilidades no funcionamento da ferramenta.

Está em estudo a viabilidade de a ferramenta ser disponibilizada com um serviço próprio, utilizando um serviço *Web portable*. Neste caso o serviço seria customizado, alterando-se as portas e também o usuário que irá disponibilizar o serviço.

Referências Bibliográficas

CAMARGO, H. A. *Automação de Tarefas*. 1. ed. Lavras-MG: UFLA - FAEPE, 2005. 83-84 p.

COSA, E. A. Controle de acesso atraves do squid. *Curso ARL - DCC / UFLA*, 2007.

DansGuardian Organization. *Site*. [S.l.], 2009. Acessado em 10 de julho de 2010. Disponível em: <http://dansguardian.org>.

DELISLE, M. *Mastering phpMyAdmin 3.1 for Effective MySQL Management*. 1. ed. Québec - Canadá: Packt, 2004.

MARCELO, A. *Squid - Configurando o Proxy para Linux*. 1. ed. Rio de Janeiro - RJ: Brasport, 2006.

MORIMOTO, C. E. *Servidores Linux*. 1. ed. São Paulo: GDH Press e Sul Editores, 2008.

PHPLDAPADMIN.ORG. *Site*. [S.l.], 2010. Acessado em 20 de fevereiro de 2010. Disponível em: <http://phpldapadmin.org>.

PHPMYADMIN.ORG. *Documentation for phpMyAdmin*. [S.l.], 2010. Acessado em 10 de fevereiro de 2010. Disponível em: <http://phplmyadmin.org>.

RUFINO, N. d. O. *Técnicas e Ferramentas de Ataque e Defesa à Redes de Computadores*. 1. ed. São Paulo-MG: Novatec, 2002.

SILVA, R. P. G.; AUGUSTO, R. B. E. *Proposta de Controle Eficaz do Acesso à Internet*. 1. ed. Vitória-ES: Faculdade Salesiana de Vitória, 2007. 42-45 p.

SIQUEIRA, L. A. *Certificação LPI - 2*. 3. ed. São Paulo: Linux New Media do Brasil, 2009. 175-176 p.

SQUID. *Squid config files*. [S.l.], 2010. Acessado em 22 de agosto de 2010. Disponível em: <http://www.squid-cache.org/Doc/config/>.

TRIGO, C. H. *OpenLDAP uma arbodagem integrada*. 1. ed. São Paulo: Novatec, 2007. 124-137 p.

UCHÔA, J. Q. *Segurança Computacional*. 2. ed. Lavras-MG: UFLA, 2005.

UCHÔA, J. Q.; SICA, F. C.; SIMEONE, L. E. *Administração em redes Linux*. 1. ed. Lavras-MG: UFLA, 2003.

WESSELS, D. *Squid: The Definitive Guide*. 1. ed. United States of America: O'Reilly Media, Inc., 2004.

Apêndice A

Arquivos de configurações

A.1 squid.conf

```
# squid.conf
# (cc) Creative Commons - Gustavo Hendrigo, Adamantina - SP - 2009

#Porta de acesso do proxy - Escuta do Dansguardian
#http_port 3128
http_port 127.0.0.1:3128

# Define o nome que irá aparecer nas páginas de erro ou acesso
# do squid
visible_hostname Servidor-Proxy

# Não faz cache de dados de formulários html, nem de resultados
# de programas cgi
hierarchy_stoplist cgi-bin ?

# Cria uma accss control list, baseando-se na url e utilizando
# expressões regulares
# nesta situaçõ foi criado uma exp. regular para cgi e ?.
acl QUERY urlpath_regex cgi-bin \?

# Não faz cache da acl QUERY
cache deny QUERY

# apache
acl apache rep_header Server ^Apache
```

```
broken_vary_encoding allow apache

# Configura o número maximo de tentativas de
# conexões em um servidor que tenha somente um endereço
maximum_single_addr_tries 1

# Tamanho máximo de memória para cache
cache_mem 56 MB

#Tamanho máximo de um objeto
maximum_object_size 20128 KB
#Com o cache_swap_high você define qual a porcentagem
# máxima que o cache deverá
#atingir para começar a apagar arquivos antigos.
# O cache_swap_low define qual a
#porcentagem deverá ser atingida durante a remoção
# desses arquivos.
cache_swap_low 80
cache_swap_high 95

#Tamanho máximo de um objeto na memória ram, caso o objeto seja
#maior que o valor estipulado ele será gravado direto no disco
maximum_object_size_in_memory 128 KB

#Define a localização do cache de disco, tamanho
#Quantidade de diretórios pai, e por fim a quantidade
# de diretórios filho
cache_dir ufs /var/spool/squid 56 32 56

#Arquivo de Log
access_log /var/log/squid/access.log squid

# Arquivo que contém os nomes de máquinas
hosts_file /etc/hosts

# Autenticação no LDAP
auth_param basic program /usr/lib/squid/ldap_auth -v 3 -b
"ou=Users,dc=phpdansadmin,dc=com,dc=br" -f "uid=%s" -h 192.168.56.2

auth_param basic children 8
auth_param basic realm Proxy. Digite seu login e senha.
auth_param basic credentialsttl 20 minutes
acl usuarios proxy_auth REQUIRED
```

```
# Tempo de atualização dos objetos relacionados aos
# protocolos ftp, gopher e http.
# Default Sugerido:
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern .               0       20%     4320

# Mínimo de Access Control List para o squid funcionar
# Não alterar estas acls, pois poderá travar o squid
# Configuração mínima
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT

#ACLs
# limita conexoes HTTP
acl connect_abertas maxconn 8

#Repasse dos ips dos clientes pelo DansGuardian
follow_x_forwarded_for allow localhost

#Acessos dos usuários autenticados.
http_access allow usuarios

#Default
```

```
http_access deny manager
http_access allow purge localhost
http_access deny purge
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

http_access allow localhost
http_access deny all
http_reply_access allow all
icp_access allow all
cache_effective_group proxy
coredump_dir /var/spool/squid
```

A.2 dansguardian.conf

```
# DansGuardian, arquivo de configuração para a versão 2.9.9.4
# VERSÃO ILUSTRATIVA COM AS PRINCIPAIS CONFIGURAÇÕES
# (NÃO COPIAR E COLAR TODO SEU CONTEÚDO)
# ** NOTA ** a partir da versão 2.7.5 a maioria da lista de
# arquivos agora estão no dansguardianf1.conf
# UNCONFIGURED - Comentar esta linha apos a configuração

# Relatório de acessos (não afeta o log)
#
#-1 = sem bloqueios - Modo Stealth
# 0 = apenas dizer 'Access Denied'
# 1 = relatar se negou frase
# 2 = relatório completo
# 3 = usar HTML arquivo de modelo (recomendado)
#
reportinglevel = 3

# Language, onde as línguas são armazenados para a internacionalização.
# O modelo HTML dentro desta dir é usado somente quando reportinglevel
# Está definido para 3. Quando usado, o DansGuardian irá exibir o arquivo
# HTML em vez de usando o script perl cgi. Esta opção é mais rápida, e mais
#fácil de personalizar a página de acesso negado.
#
languagedir = '/etc/dansguardian/languages'

# language, utilizar a partir de languagedir.
language = 'portuguese'
```

```
# Configurações de Logs
#
# 0 = nenhum 1 = apenas negado 2 = negou todo o texto base 3 = todas solicitações
loglevel = 2

# the page gets let through.  Can be useful for diagnosing
# why a site gets through the filter.
# 0 = never log exceptions
# 1 = log exceptions, but do not explicitly mark them as such
# 2 = always log & mark exceptions (default)

# Log se uma exceção (usuário, IP, URL, frase) é correspondida e assim
# A página obtém permissão para passar. Pode ser útil para diagnosticar
# porque um site é liberado no filtro.
# 0 = sem log excepções
# 1 = exceções log, mas não explicitamente marcá-los como tal
# 2 = log sempre marca exceções (default)

logexceptionhits = 2

# Formato do log
# 1 = formato DansGuardian
# 2 = formato CSV
# 3 = Formato de log do Squid
# 4 = delimitado por tabulações

logfileformat = 1

# Local dos arquivos de log
# Define o diretório e o nome do arquivo de log.

loglocation = '/var/log/dansguardian/access.log'

# Configurações de rede

# IP que DansGuardian escuta. Se for deixado em branco DansGuardian
# escuta todos. Isso inclui todas as NICs, loopback, modem, etc. Normalmente,
# um firewall protegeria isso, é possível limitar a um determinado IP
# ou especificar cada IP em uma linha de filterip individual.
filterip =

# A porta que DansGuardian escuta.
filterport = 8080
```

```
# O ip do proxy (o padrão é o loopback - ou seja, esse servidor)
proxyip = 127.0.0.1

# Porta ond o DansGuardian se conecta
proxyport = 3128

# Opções de filtors dos grupos
# filtergroups, define o número de grupos de filtro. Um grupo é um
# conjunto de filtro de conteúdo opções de filtragem será aplicado
# a um grupo de usuários. O valor deve ser 1 ou maior.
# O DansGuardian irá procurar automaticamente por dansguardianfN.conf
# onde N é o filtro grupo. Para atribuir usuários a grupos de usar a
# opção filtergroupslist. Todos os sem grupos definidos são do grupo 1.
# Deve-se ter algum tipo de autenticação para ser capaz de usuários ser
# atribuidos a um grupo. Quantos mais grupos, maior necessidade de memória RAM
# Use o mínimo possível.

filtergroups = 2
filtergroupslist = '/etc/dansguardian/lists/filtergroupslist'

# Arquivos de autenticação local
bannediplist = '/etc/dansguardian/lists/bannediplist'
exceptioniplist = '/etc/dansguardian/lists/exceptioniplist'

# Modo de frase ponderada
# Existem três modos de operação possíveis:
# 0 = off = não usar o recurso de frase ponderada.
# 1 = a, = normal funcionamento normal frase ponderada.
# 2 = no singular, cada frase = é considerada em apenas uma soma em uma página.
#
weightedphrasemode = 2


# Bloquear ou limitar upload
# Medido em kibibytes após a codificação MIME e bumph cabeçalho
# Use 0 para o de blocos completos
# Use mais elevada (por exemplo, 512 = 512Kbytes) para limitar
# Use -1 para não bloquear
#maxuploadsize = 512
#maxuploadsize = 0
maxuploadsize = -1

# Plugins
# Estas opções substituiem o * usernameidmethod nas versões anteriores. Eles
```

```
# lidam com a extração de nomes de clientes de várias fontes, tais como
# autorização cabeçalhos Proxy e servidores de ident, permitindo que os pedidos
# sejam tratados de acordo com as definições do grupo de usuário do filtro.
# Por exemplo, se o Squid for com a autenticação NTLM e autenticação básica
# ativada e tanto o "proxy-base e 'ntlm proxy' auth plugins são ativados aqui,
#
# Se não usar grupos de filtros, não é necesseário especificar essa opção.
authplugin = '/ etc / dansguardian / authplugins / proxy basic.conf'
# Authplugin = '/ etc / dansguardian / authplugins / proxy digest.conf'
# Authplugin = '/ etc / dansguardian / authplugins / proxy ntlm.conf'
# Authplugin = '/ etc / dansguardian / authplugins / ident.conf'
authplugin = '/ etc / dansguardian / authplugins / ip.conf'

# Se adiciona um X-transmitido-Para: <clientip> ao pedido HTTP
# header. Isso pode ajudar a resolver alguns sites problema que precisa saber
# o ip fonte. on | off

usexforwardedfor = off

# Define o número máximo de processos
# Valor máximo 250

maxchildren = 120

# Númermo mínimo de processos
minchildren = 8

# Número mínimo de processos que devem estar prontos para lidar com conexões.
minsparechildren = 4
```

Apêndice B

Lista de Categorias

B.1 Lista de categorias da *blacklist*

```
abortion - Aborto excluindo as informações quando relacionados com religião;
ads - URLs de anúncios - servidores classificado e proibidos;
adult - Sites contendo material adulto, tais como palavrões, mas não pornô;
aggressive - Semelhante à violência, mas mais do que promover retratando;
antispyware - Sites que remove spyware;
artnudes - Sites contendo nudez artística;
astrology - Sites de Astrologia;
audio-video - Sites com audio ou video para downloads;
banking - Sites de bancos;
beerliquorinfo - Sites com informações apenas sobre a cerveja ou licores;
beerliquorsale - Sites com cerveja ou licores à venda;
blog - Blogs;
cellphones - Materias para telefones celulares móveis;
chat - Sites com salas de chat, etc;
childcare - Sites para crianças;
cleaning - Sites sobre limpeza;
clothing - Sites sobre e venda de roupa;
contraception - Informação sobre a contracepção;
culnary - Sites sobre culinária;
dating - Sites sobre namoro;
desktopsillies - Sites contendo protetores de tela, fundos, CURSERS,
 ponteiros. Temas e semelhantes, potencialmente perigosos conteúdo;
dialers - Sites com discadores tais como os de pornografia ou trojans;
drugs - Sites relacionados a drogas e medicamentos;
ecommerce - Sites que fornecem compras on-line;
```

entertainment – Sites que promovem filmes, livros, revistas, humor;
filehosting – Sites servidores de arquivos;
frencheducation – Sites sobre educação francesa;
gambling – Pornografia, heróticos;
games – Sites relacionados a jogos;
gardening – Sites sobre jardinagem;
government – Militar e Governos;
guns – Sites com armas;
hacking – Informações Hacking, cracking;
homerepair – Sites sobre reformas de casas;
hygiene – Sites sobre higiene e outros materiais de higiene pessoal;
instantmessaging – Sites que contêm o download ou cliente e messenger;
jewelry – Sites de informações e venda de jóias;
jobsearch – Sites para encontrar emprego;
kidstimewasting – Sites para perder tempo;
mail – Webmail e sites de email;
marketingware – Sites sobre a comercialização de produtos;
medical – Sites de Medicina;
mixed_adult – Sites de conteúdo adulto;
mobile-phone – Sites sobre telefones celulares;
naturism – Sites que contenham imagens de nudez e ou promover naturismo;
news – Sites de notícias;
onlineauctions – Leilões online;
onlinegames – Jogos online;
onlinepayment – Pagamentos online;
personalfinance – Locais de Finanças Pessoais;
pets – Sites sobre animais de estimação;
phishing – Sites que tenta enganar pessoas para dar informações privadas;
porn – Pornografia;
proxy – Sites com proxy para burlar filtros;
radio – Sites relacionados com rádio e televisão;
religion – Sites para promoção da religião;
ringtones – Sites contendo toques, jogos, imagens e outros;
searchengines – Sites de busca como o Google;
sect – Sites sobre seitas;
sexuality – Sites dedicados à sexualidade, possivelmente inclui material adulto;
shopping – Sites de compras;
socialnetworking – Redes Sociais como o Orkut;
sportnews – Noticias de esportes;
sports – Sites de esportes;
spyware – Sites que administram ou têm spyware para download;
updatesites – Sites de atualizações de software são baixados, incluindo antivírus;
vacation – Sites sobre onde ir em férias;
violence – Sites que contêm violência;

virusinfected – Sites hospedeiros de vírus;
warez – Sites com software pirata;
weather – Clima e sites de notícias relacionados com o clima;
weapons – Sites detalhando ou vende armas;
webmail – Sites de webmail;
whitelist – Contém site especificamente 100 po cento adequado para as crianças.