Hedwio Carvalho e Silva

Solução de controle de acesso a Internet em ambientes corporativos

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação "*Lato Sensu*" Administração em Redes Linux, para a obtenção do título de especialista.

Orientador
Prof. DSc. Gustavo Guimarães Parma

Lavras
Minas Gerais - Brasil
2005

Hedwio Carvalho e Silva

Solução de controle de acesso a Internet em ambientes corporativos

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação "*Lato Sensu*" Administração em Redes Linux, para a obtenção do título de especialista.

Aprovada em *17 de Abril de 2005*

Prof. Heitor Augustus Xavier Costa

Prof. Sandro Pereira Melo

Prof. DSc. Gustavo Guimarães Parma
(Orientador)

Lavras
Minas Gerais - Brasil

Agradecimentos

Ao Prof. Parma, pela paciência e pela orientação durante esta jornada.

Aos colegas de curso, que de forma silenciosa também são responsáveis por este trabalho, em especial aos colegas Aldemon e Paulo de Carvalho, pela ajuda nos momentos finais.

A meu pai e meus irmãos que, a distância, também foram incentivadores.

A minha mãe, que sempre me acompanha em cada importante passo em minha vida.

A minha esposa Milena e a meu filho Caio, sempre presentes.

Resumo

As corporações têm a necessidade de controlar o acesso a Internet proveniente da sua rede interna a fim de aumentar a produtividade da empresa e de dimimuir os riscos associados ao acesso irrestrito a rede mundial de computadores. Esta solução implementa controle de acesso a Internet através de ferramentas que permitem validação dos usuarios na rede sem a necessidade de nova autenticação no serviço de *proxy*. Filtros de conteúdo também são aplicados juntamente com ferramenta de geração de relatórios estatísticos de acesso.

Sumário

Lista de Figuras

Capítulo 1

Introdução

A necessidade de controlar o acesso a *Internet* proveniente da rede interna com o objetivo de aumentar a produtividade da empresa e de dimimuir os riscos associados ao acesso irrestrito a rede mundial de computadores motivou a implementação desta solução baseada em *software livre*.

Este trabalho é o resultado da compilação de um conjunto de ferramentas disponíveis na Internet com o objetivo de fornecer ao leitor o entendimento necessário que permita a implementação de uma solução de controle de acesso a Internet com filtros de conteúdo e geração de relatórios de uso em ambientes corporativos através de softwares livres. Os serviços de *proxy*, *cache*, filtros de conteúdo, validação de autenticação de usuários possibitam tal implementação.

O uso do serviço de *proxy* permite concentrar todo o fluxo de acesso a Internet através de um único equipamento, o servidor *Proxy*. Este servidor, a fim de melhorar a performance, também implementa o serviço de cache, guardando localmente partes dos sítios acessados, e retornando esses conteúdos ao usuário de forma muito mais rápida do que uma nova consulta ao sítio.

Contudo, faz-se necessário a implementação de um serviço de filtro de conteúdo para se obter, efetivamente, um controle no acesso a sítios considerados indevidos através de análise do conteúdo destes sítios. Outra ferramenta importante, principalmente para o corpo gerencial da corporação, diz respeito a geração de relatórios dos acessos feitos pelos funcionários da instituição.

As redes em funcionamento nas instituições atualmente já implementam um esquema de autenticação de usuários, quer seja em controladores de domínios proprietários, quer baseados em software livre. Tais esquemas de autenticação devem ser incorporados a esta solução de controle de acesso a Internet sem a necessidade do usuário autenticar-se novamente, fazendo assim a validação da autenticação do usuário. Vários esquemas de autenticação são suportados pelas ferramentas utili-

zadas dentre os quais pode-se destacar: *LDAP*, *NTLM*, *PAM*, *SMB* e *winbind*, que serão alvo de estudo no decorrer deste trabalho.

As ferramentas utilizadas são bastante conhecidas e largamente utilizadas em ambientes de redes de computadores em diversos lugares espalhados pelo mundo e são licenciadas sob a *GPL*:

- Sistema operacional linux;
- *Squid*, que se propõe a funcionar como *proxy* e *cache* para os protocolos *HTTP*, *HTTPS*, *FTP* e *GOPHER*;
- *Dansguardian*[1], que efetua filtros de conteúdo;
- *SARG*, que é uma ferramenta de geração de relatórios de logs do *Squid* e permite ao administrador identificar os sítios acessados pelos usuários na Internet;
- *SAMBA*, que permite validação da autenticacão dos usuários no controlador de domínios;
- *OpenLDAP*, que permite acesso ao serviço de diretórios em execução através do protocolo *LDAP*;
- *PAM*, que através de seus vários módulos possibilita um controle referente a autenticação de usuários.

[1] Para maiores informações sobre licenciamento ver http://dansguardian.org/?page=copyright2

Capítulo 2

Conceitos Básicos

Alguns conceitos são necessários para o entendimento deste texto. Assim, com o objetivo de contextualizar o leitor, este Capítulo trata de algumas definições e explicações a respeito de Controle de Acesso, Filtro de Conteúdo, *Proxy/Cache*, Controlador de Domínio e Autenticação de usuários.

2.1 Controle de Acesso

O controle no acesso a Internet proveniente de redes privativas é uma tema bastante controverso. Algumas pessoas insistem em dizer que os profissionais contratados por uma empresa tem direito a privacidade no acesso a Internet. Sem levar em consideração que um contrato de trabalho foi assinado e que as atividades são exercidas em nome da empresa em seu ambiente físico e utilizando-se de recursos da empresa.

Contudo, a alegação das empresas é bastante convincente: minimizar os riscos de vírus, *trojans* e outras pragas da Internet e aumentar a produtividade. A fim de obterem uma base legal, as corporações têm adotado fortemente a estratégia de embasarem-se em uma política de segurança ou em um termo de compromisso assinado pelo funcionário quando de sua entrada na empresa em que toma conhecimento e concorda com o procedimento.

Para, efetivamente, controlar o acesso a Internet, são utilizadas várias ferramentas que permitem identificar que usuário e/ou máquina da rede acessou determinado sítio em determinada data e hora, gerando assim informações que podem ser utilizadas pela gerência para tomar as devidas providências.

Pesquisas realizadas nos últimos anos têm demonstrado pelo tipo de sítio acessado que a produtividade dos funcionários que têm acesso irrestrito a Internet em ambiente de trabalho é baixa.

Segundo o documento *"Internet Filtering Alternatives White Paper"*(SOFTWARE, 2003), quando trata de estatísticas de abuso na Internet:

- Acesso a sítios de sexo foi reportado por 62% das organizações (*PC Week*);
- Acesso a Internet em ambiente de trabalho gera cerca de 30 a 40% de queda de produtividade (*IDC Research*);
- Cerca de 70% de todo o tráfego pornográfico na Internet ocorre no horário de 9h as 17h (*SexTracker*);
- 32,6% dos empregados não tem objetivo específico quando acessam a Internet (*eMarketer.com*);
- Um em cada cinco homens e uma em cada oito mulheres admitiram utilizar seus computadores do trabalho como principal equipamento em que acessam conteúdos ligados a sexo (*MSNBC*);
- Usuários de Internet no escritório utilizam-se da vantagem de conexões em alta velocidade para acessar sítios de entretenimento como *broadcast.com* e *mp3.com* mais frequentemente que em suas casas (*Nielsen/Net Ratings*);
- 82% dos executivos de negócios dos Estados Unidos consultados pela empresa de consultoria *Dataquest* (uma divisão do *Gartner Group*) acreditam que o uso da Internet deveria ser monitorado em suas companhias (*InformationWeek Online*).

2.2 Filtro de Conteúdo

O serviço de *Proxy* funciona muito bem para atender às necessidades para as quais foi criado: intermediar requisições de clientes com destino a servidores na Internet, além de efetuar filtros baseados em listas de controle de acesso capazes de bloquear o acesso a determinados sítios.

Contudo, no que diz respeito a proibição de acesso a sítios na Internet, uma característica bastante importante não pode ser implementada através de servidores *proxies* exclusivamente. A capacidade de proibir o acesso de acordo com o conteúdo de cada página acessada. Tal necessidade é suprida por ferramentas conhecidas como filtros de conteúdo.

Essas ferramentas percorrem cada página acessada antes de disponibilizá-la ao usuário e efetuam uma verificação de termos, palavras e frases consideradas inadequadas segundo base de dados de tais conteúdos. A maioria dessas aplicações

permite ao administrador informar dentre o conteúdo dessas listas o que, efetivamente, será considerado termo indevido. Além de permitir acréscimo de conteúdo a serem bloqueados.

Ainda, segundo *"Internet Filtering Alternatives White Paper"* (SOFTWARE, 2003), os filtros de conteúdo podem ser divididos nos seguintes tipos:

***Appliances* dedicados a filtragem** Esta categoria inclui dispositivos especificamente projetados para controle e monitoração do acesso a Internet;

Servidores pré-configurados para filtragem Esta categoria inclui soluções onde os equipamentos servidores de propósito geral são configurados de fábrica com aplicação de filtragem de conteúdo;

Aplicação de filtragem baseada em servidor Esta categoria inclui programas que trabalham com sistemas operacionais em plataforma Windows NT ou Unix, ou como um *plugin* para aplicações de servidores *proxy*;

Addons* para serviço de filtragem nos *firewalls Serviços que podem ser adicionados ao *firewall* ou a outros dispositivos de rede;

Aplicação de filtragem baseada no cliente Esta categoria inclui aplicações para *Windows* e *Macintosh* e plugins de navegadores que são adicionados as tais características no computador do usuário.

De uma forma mais específica, pode-se definir o filtro de conteúdo baseado em pesos com uma nova categoria. Assim, é possível informar a aplicação que o simples aparecimento de uma determinada palavra em um sítio não é suficiente para bloqueá-lo. Mas, um outro termo pode ser considerado indevido somente no caso de aparecer várias vezes no mesmo sítio.

2.3 *Proxy*/*Cache*

O contínuo avanço tecnológico tem permitido às empresas experimentarem velocidades cada vez maiores no acesso a Internet. Porém, as necessidades dessas instituições também crescem diariamente. Assim, mesmo com um acesso considerado rápido, acréscimos de velocidade, que geram diminuição no tempo de resposta, são sempre bem-vindos. O serviço de *cache* tem esse propósito: ganho de performance.

O serviço de *Proxy* por sua vez tem o papel de concentrar todas as requisições das mais diversas origens, canalizando-as por uma mesma saida. Ele é que, efetivamente, faz a requisição ao destino. Funciona como um intermediário entre o

cliente e o servidor de destino. Esse intermediário efetua tais requisições segundo regras, ou filtros, implementados pela ferramenta de *Proxy*. Tais filtros têm a função de proibir ou liberar acessos a sítios, endereços identificadores de máquinas e redes, *strings* e até limitar velocidade de acesso. Além de ser capaz de coibir o acesso através de regras que atuam sobre os clientes da rede interna: nomes de usuários, grupos de usuários, endereços identificadores de máquinas, etc.

Soluções presentes no mercado contemplam a utilização dos serviços de *Proxy* e *Cache* juntos. Sendo que o tratamento da informação entre tais serviços é realizado automaticamente pela aplicação sem a necessidade de intervenção do administrador, conforme a Figura 2.1.

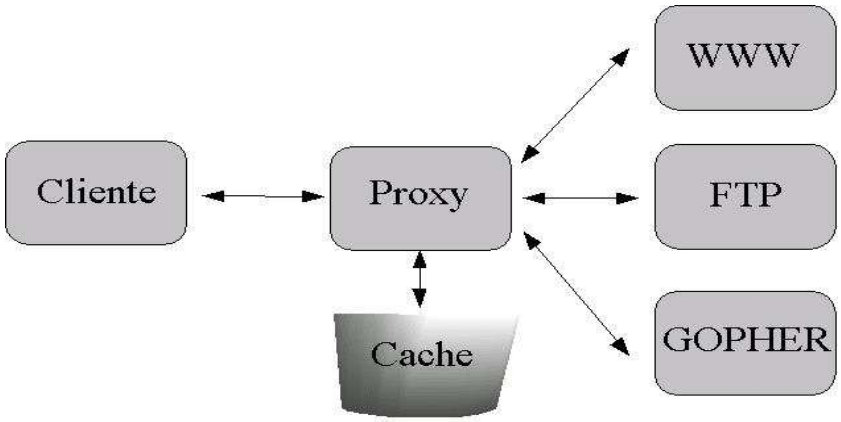

Figura 2.1: Funcionamento de *Proxy* e *Cache*.

2.4 Controlador de Domínio

O uso de redes de computadores em ambientes de trabalho tem aumentado a cada dia. Isto é facilmente percebido ao analisar-se pequenas empresas que não têm a informática como área fim. Essas empresas, que até bem pouco tempo utilizavam-se de máquinas de datilografia, renderam-se a tecnologia. E, hoje, é muito difícil uma micro-empresa que não faça uso de computadores no seu dia-a-dia.

Mesmo em ambientes mais restritos, com poucos computadores, tem-se percebido que a utilização de *redes ponto-a-ponto* não se adequa muito bem, uma vez que as informações associadas as atividades da empresa não estão localizadas

em ponto central, com todos os cuidados necessários: backup das informações, controle de acesso, garantia de disponibilidade, dentre outros.

A fim de prover um ambiente de rede mais adequado às atividades da corporação, além de equipamentos servidores que mantenham a guarda das informações, faz-se necessário a implementação de uma outra solução capaz de autorizar e autenticar os funcionários na rede privativa da instituição: o controlador de domínio.

A garantia de acesso dos usuários ao ambiente de rede da empresa que permitirá associar as ações realizadas na rede ao usuário que, efetivamente, a executou também é papel do controlador de domínio.

Existem algumas soluções de controladores de domínio disponíveis no mercado. A maioria delas proprietária. Contudo, soluções baseadas em *software livre* também estão disponíveis para uso e serão consideradas neste trabalho.

2.5 Autenticação de Usuários

A fim de obter acesso a uma rede, o usuário solicita sua autenticação informando nome na rede e a senha de acesso. O sistema efetua a autenticação verificando se nome de usuário e senha informados estão realmente associados àquele usuário específico e estão corretos. Quando um usuário obtém acesso a rede, ele foi autenticado.

Porém, inúmeras aplicações utilizam-se do conceito de autenticação de usuários para seu funcionamento. Algumas delas utilizam módulos adicionais para efetivar a autenticação através de diferentes protocolos como *Kerberos*, *LDAP* e *NTLM*. Outras aplicações apenas validam a autenticação efetuada anteriormente garantindo os acessos do usuário.

Para que a validação do usuário autenticado funcione a contento, algumas ferramentas adicionais devem ser implementadas, algumas na própria aplicação e outras nos servidores que executam as aplicações.

Existem várias opções de ferramentas para validação de usuários que podem ser implementadas no servidor em que a aplicação é executada a fim de adequar a solução de autenticação de usuários em produção. Várias dessas ferramentas estão sendo muito utilizadas atualmente com sucesso, como: *PAM*, *OpenLDAP* e *Winbind*.

Capítulo 3

Implementação da Solução de Integração de Ferramentas

Este Capítulo mostra a configuração das ferramentas utilizadas, não esquecendo de definir e detalhar o funcionamento de cada uma delas.

Também é conteúdo deste Capítulo a validação de autenticação de usuários em um controlador de domínio por tratar-se de uma necessidade prática encontrada nos ambientes corporativos.

Por fim, trata-se da integração das ferramentas selecionadas com o objetivo de, efetivamente, controlar o acesso a Internet dos usuários da rede corporativa.

3.1 Ferramentas Utilizadas

Esta Seção trata de questões relativas as ferramentas utilizadas para implementar a solução de controle de acesso a Internet. São consideradas desde questões de Definição de cada ferramenta até Funcionamento, Características Adicionais e Conclusões as quais se chegou após os testes e a implementação.

3.1.1 *Squid*

Definição

O Squid[1] nasceu do projeto de um servidor *HTTP* que também incorporava *Proxy* e *Cache* na década de 90. Tendo sido desenvolvido pelo *Internet Research Task Force Group on Resource Discovery (IRTF-RD)* através do projeto *Harvest*. E,

[1] ver http://www.squid-cache.org

atualmente, vem sendo melhorado por um grupo considerável de desenvolvedores(WESSELS, 2004).

Segundo (WESSELS, 2004), trata-se de uma ferramenta capaz de aceitar requisições *HTTP* e *HTTPS* de clientes e capaz de efetuar requisições *HTTP*, *FTP* e *Gopher* para servidores, além de implementar várias características comumente úteis em ambientes corporativos:

- Controle de banda no acesso a Internet;
- Redução do tempo de carga de páginas na Internet;
- Coleta de estatísticas do tráfego de acesso a Internet proveniente da rede privativa;
- Bloqueio de sítios considerados de conteúdo inapropriado;
- Garantia de que somente os usuários autorizados terão acesso a Internet;
- Conversão de requisições *HTTPS* de um lado em *HTTP* do outro lado;
- Proteção de máquinas internas de acessos externos uma vez que as requisições a sítios externos são efetuadas pelo *Proxy*.

Funcionamento

O *Squid* foi escrito com a preocupação de ser portável, assim ele funciona na maioria dos sistemas operacionais *Unix*, como: *Linux, BSD/OS, FreeBSD, NetBSD, OpenBSD, Solaris, HP-UX, OSF/DUNIX/TRU-64, Mac OS/X, IRIX e AIX*, além de funcionar em ambientes *Microsoft Windows*(WESSELS, 2004).

Os requisitos de *hardware* necessários para sua implementação são, em geral, modestos. Mesmo assim, memória é o recurso mais importante, uma vez que pouca quantidade de memória degrada consideravelmente a performance. Espaço em disco é um outro fator importante, pois mais espaço em disco significa mais objetos em *cache* e, portanto, menores tempos de resposta.

O fato de ser *Proxy* permite ao *Squid* intermediar as transações entre clientes e servidores. Ele aceita requisições dos clientes, processa e as encaminha ao servidor desejado. Tais requisições podem ser registradas, rejeitadas e modificadas antes do encaminhamento.

Por funcionar como *Cache*, a ferramenta armazena localmente conteúdo de páginas acessadas recentemente com o objetivo de reutilizá-las, aumentando assim a performance pela diminuição do tempo de resposta.

A característica de *Cache* é passível de desabilitação, o que não ocorre com a função de *Proxy*, por ser a essência do *Squid*.

Características adicionais

A cada nova versão o *Squid* tem crescido em tamanho e funcionalidade.

Uma característica bastante interessante da aplicação é a implementação de várias funcionalidades através do uso de *ACLs (Access Control List)*.

Esta implementação agrega um poder fabuloso ao software pois permite a criação de listas capazes de filtrar desde simples domínios até tipos de conteúdo especificados (*mime types*).

Outros pontos fortes são: numerosos módulos de autenticação, desde *NCSA* até autenticação baseada em *LDAP* ou *Kerberos*, e avançadas opções de armazenamento de disco e interceptação *HTTP*.

Conclusões

Por fim, pode-se referendar o Squid como uma ferramenta poderosa e flexível, capaz de ser implementada em ambientes corporativos sem maiores receios.

3.1.2 *Dansguardian*

Definição

O Dansguardian[2] é uma ferramenta capaz de filtrar acessos a Internet com base em diferentes critérios (BARRON, 2003b):

- Filtros de domínios ou *URLs* com uma performance visivelmente superior a outras ferramentas;
- Filtros baseados em frases associadas a pornografia ou consideradas inapropriadas;
- Filtros por figuras (*PICS*) ou por tipo de conteúdos (*MIME*);
- Filtros por extensão de arquivos, como: .exe, .dll, .scr e outros;
- Filtros do tipo *POST*, em que é possível bloquear ou limitar *upload* na Internet.

A ferramenta difere da maioria disponível no mercado pelo fato de não funcionar apenas como filtro de *URL*, mas também como um efetivo filtro de conteúdos de páginas *Web*. Pois, faz uma varredura do conteúdo de cada página acessada por seus usuários e não somente uma liberação ou proibição do nome do sítio ou da URL acessada.

[2] ver http://www.dansguardian.org

Funcionamento

Este filtro de conteúdo funciona em conjunto com qualquer *Proxy*, podendo ser instalado em sistemas operacionais *Linux*, *FreeBSD*, *OpenBSD*, *NetBSD*, *Mac OS X*, *HP-UX*, e *Solaris* (BARRON, 2003b).

O Dansguardian não tem características de *Proxy*, portanto é obrigatório o uso de um servidor *Proxy* para que a ferramenta seja implementada.

Nas soluções comumente encontradas no mercado, o filtro de conteúdo recebe as requisições do navegador do usuário, aplica as restrições estabelecidas ou as exceções configuradas e, em seguida, passa a requisição para o *Proxy* conforme Figura 3.1. Este faz o seu papel: a intermediação entre o cliente e o servidor a ser acessado.

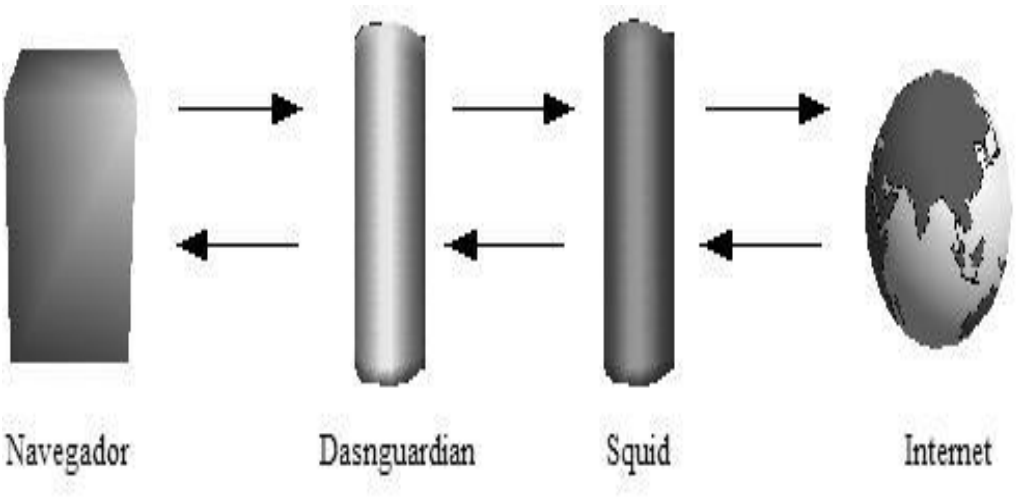

Figura 3.1: Fluxo da requisição do cliente via Filtro de Conteúdo.

No processamento interno de arquivos contendo proibições e exceções, existe uma ordem pré-estabelecida conforme Tabela 3.1.

Assim, é importante que o administrador conheça tal fluxo de processamento a fim de que suas configurações funcionem a contento.

Tabela 3.1: Ordem de processamento dos filtros no Dansguardian

Ordem de Processamento	Arquivo
1	exceptioniplist
2	exceptionuserlist
3	exceptionsitelist
4	exceptionurllist
5	blanket block
6	bannediplist
7	banneduserlist
8	bannedregexpurllist
9	bannedurllist
10	blacket ip block
11	bannedsitelist
12	postupload
13	bannedmimetypelist
14	bannedextensionlist
15	exceptionphraselist
16	bannedphraselist
17	weightedphraselist

Características adicionais

O algoritmo de checagem de frases proibidas é bem inteligente, sendo capaz de verificar códigos HTML e espaços em branco.

Contém um modo capaz de efetuar apenas o registro dos sítios acessados sem efetivar o bloqueio, podendo ser feito sem o conhecimento do usuário.

Os registros produzidos tem formato legível aos olhos humanos, os quais podem ser gerados em formato *CSV* a fim de ser exportado para bancos de dados (BARRON, 2003b), além de permitir o registro por nome de usuário autenticado.

O *Dansguardian* permite filtragem de *URL* baseado em expressões regulares ou de requisições *HTTPS*.

Uma última funcionalidade considerável é a capacidade de implementar tais filtros através de um esquema de ponderação em que cada termo, frase, sítio ou *URL* tem um peso associado e o bloqueio pode se dar através da soma dos pesos encontrados. Vale lembrar que existem pesos com valores positivos e negativos.

Os pesos com valores positivos são somados para gerar um valor total e implementar um filtro, enquanto os pesos com valores negativos quando somados diminuirão o peso total. Assim, é possível liberar determinados acessos através do acréscimo de um filtro com peso de valores negativos.

Na prática, tal funcionalidade implementa filtros mais condizentes com a verdade.

Conclusões

Após intensa pesquisa e realização de vários testes de funcionalidade e de performance, notou-se alguns diferenciais do *Dansguardian* em relação a outras ferramentas licenciadas sob *GPL* capazes de realizar filtros no acesso a Internet. Em relação aos testes de funcionalidade, não foi encontrada outra solução com efetivo filtro de conteúdo, mas somente com filtro de *URLs*. Quanto aos testes de performance, foi verificado que o algoritmo utilizado no *Dansguardian* é bem mais rápido que as outras ferramentas como o *SquidGuard*[3].

3.1.3 *Samba*

Definição

Os ambientes de redes de computadores atualmente têm uma variedade enorme de sistemas operacionais; *UNIX*, *Windows*, *Linux* são exemplos. Essa heterogenei-

[3]ver http://squidguard.org

dade faz os serviços de compartilhamento de arquivos e impressoras e de autenticação de usuários contra domínios tornar-se algo complexo.

O *Samba* é um conjunto de aplicações *Open Source* licenciado sob *GPL* que provê a funcionalidade *CIFS (Common Internet File System)/ SMB (Server Message Block)* em ambientes *UNIX*, tendo se mostrado como uma excelente ferramenta para administração de tais ambientes heterogêneos.

Esse conjunto de aplicações permite aos usuários *Windows* acessarem arquivos no *Linux* e vice-versa. Além de emular características de compartilhamento de arquivos nativa do *Windows*.

Um equipamento com *Samba* instalado pode se mostrar como um servidor *Microsoft* e prover serviços como (TERPSTRA, 2004):

- Compartilhamento de um ou mais sistemas de arquivos;
- Compartilhamento de impressoras tanto no servidor quanto nos clientes;
- Uso de ferramentas de visualização de estações na rede;
- Autenticação de clientes contra um domínio *Windows* configurado no servidor *Samba*;
- Funcionamento como um servidor de nomes *WINS (Windows Internet Name Service)*;

Funcionamento

Dois programas principais compõem o *Samba*: o *smbd* e o *nmbd*.

O primeiro é responsável pelos de serviços de compartilhamento e de impressão. Provendo autorização e autenticação através dos modos *user* e *share*. Esses modos são utilizados para proteger os serviços de compartilhamento e impressão através do uso de senhas. No caso do modo *share*, uma senha é dada a qualquer usuário que possa acessar o compartilhamento. No modo *user*, a autenticação se dá por usuário através de nome de usuário e senha.

O segundo está envolvido com o gerenciamento e a distribuição de listas de nomes *NetBIOS* (TERPSTRA, 2004)[4]

O pacote *Samba* contém outras ferramentas úteis em ambientes de redes heterogêneos, a saber:

[4]Network Basic Input Output System é um pedaço de programa carregado na memória para prover uma interface entre programas e hardwares de rede usado em ambientes Microsoft. Ele inclui um esquema de endereçamento usando nomes de 16 bytes para identificar estações de trabalho e aplicações na rede.

- *smbclient*, cliente *FTP-like Unix* que pode ser usado para conectar a compartilhamentos *Samba*;

- *smbtar*, programa para efetuar *backups* em compartilhamentos no estilo do comando *tar* do *Unix*;

- *nmblookup*, programa que provê resolução de nomes *NetBIOS* sobre *TCP/IP*;

- *smbpasswd*, ferramenta que permite ao administrador alterar senhas criptografadas usadas pelo *Samba*;

- *smbstatus*, programa que mostra as conexões de rede ativas nos compartilhamentos no servidor *Samba*;

- *testparm*, aplicação simples que valida o arquivo de configuração do *Samba*;

- *testprns*, programa que testa quando várias impressoras são reconhecidas pelo *smbd*.

Características adicionais

A versão 3 do *Samba* acrescenta características importantes para os ambientes de rede encontrados atualmente, com tecnologias novas que estão sendo utilizadas nas corporações (TERPSTRA, 2004), conforme itens abaixo:

- Suporte ao *Active Directory*, com permissão de cadastramento como *servidor membro* do domínio *Windows* e autenticação de usuários através de *LDAP* e *Kerberos*;

- Melhoria do suporte a impressão incluindo publicação de atributos de impressora no *Active Directory*;

- Suporte para migração de domínio *Windows NT 4.0* para domínio Samba com manutenção de *SID* de usuário, grupo e domínio;

- Suporte para estabelecimento de relação de confiança com controladores de domínio *Windows NT 4.0*;

- Suporte completo para cliente e servidor *SMB* com garantia de compatibilidade com as características de segurança padrão do *Windows 2003*.

Conclusões

Trata-se de ferramenta reconhecidamente poderosa utilizada em larga escala quando existe a necessidade de compartilhamento de arquivos e impressoras e/ou autenticação de usuários em redes compostas por sistemas operacionais diferentes entre clientes e servidores. Particularmente no que diz respeito a integração entre clientes *Linux* e servidores *Windows*, ou vice-versa, o conjunto de aplicações *Samba* agrega facilidades extremamente importantes em redes mistas.

3.1.4 *OpenLDAP*

Definição

LDAP (Lightweight Directory Access Protocol) é um protocolo cliente-servidor, que funciona na pilha *TCP/IP*, utilizado para acessar um serviço de Diretório. Ele foi inicialmente usado como uma interface para o *X.500*, mas também pode ser usado com autonomia e com outros tipos de servidores de Diretório.

Tecnicamente, *LDAP* é um protocolo de acesso a diretório para um serviço de diretório *X.500*, serviço de diretório do modelo *OSI*. Inicialmente, clientes *LDAP* acessavam *gateways* para serviços de diretório *X.500*. Este *gateway* funcionava entre o cliente e o *DAP (Directory Access Protocol)* e entre o *gateway* e o servidor *X.500*. O *DAP (Directory Access Protocol)* é um protocolo *pesado* que opera sobre a pilha de protocolos do modelo *OSI* (*Open Systems Interconnection*) e requer uma quantidade significante de recursos de computação. O *LDAP* foi projetado para operar sobre a pilha *TCP/IP* e prover uma maior funcionalidade em relação ao *DAP* a um custo muito menor (FOUNDATION, 2004).

Principais Características

O modelo da informação *LDAP* é baseado em entradas. Uma entrada é uma coleção de atributos que tem um identificador único, *Distinguished Name (DN)*. O *DN* é usado para refenciar entradas de forma a não existirem ambiguidades. Cada atributo de uma entrada tem um tipo e um ou mais valores. Os tipos são tipicamente *strings* mnemônicas, como *cn* para *common name* ou *mail* para endereços de email. A sintaxe dos valores dependem do tipo de atributo. Assim, um atributo *mail* pode conter valores como *ze@minhaempresa.com*. Um atributo *jpegphoto* poderia conter um fotografia em um formato *JPEG*.

No *LDAP*, entradas de diretórios são organizadas em forma de árvore. Tradicionalmemte, esta estrutura reflete uma hierarquia geográfica ou organizacional. Entradas representando países no topo da árvore. Abaixo, as entradas representando os estados e as organizações nacionais. Mais abaixo, entradas representando

unidades organizacionais, pessoas, impressoras, documentos ou uma outra informação (FOUNDATION, 2004).

A árvore LDAP também pode ser organizada por nomes de domínios Internet. Esta organização é mais popular e pode ser vista na Figura 3.2.

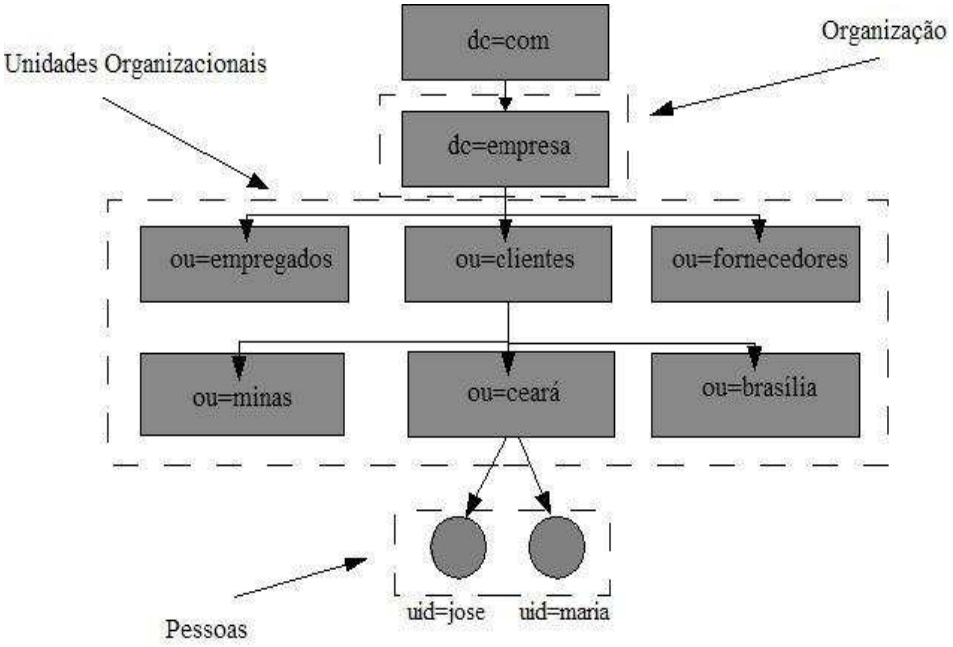

Figura 3.2: Árvore *LDAP*.

Atualmente o *LDAP* vem se tornando um padrão e diversos programas têm suporte a este protocolo. Livros de endereços, autenticação e armazenamento de certificados digitais (S/MIME) e de chaves públicas (PGP) são alguns dos exemplos onde o *LDAP* é amplamente utilizado (SZTOLTZ, 2003).

Funcionamento

Um Diretório é como um banco de dados, mas tende a conter mais informações descritivas, baseadas em atributos e é organizado em forma de árvore, não de tabela. A informação em um Diretório é geralmente mais lida do que é escrita. Como conseqüência, Diretórios normalmente não são usados para implementar transações complexas, ou esquemas de consultas regulares em bancos de dados, transações estas que são usadas para fazer um grande volume de atualizações complexas. Atualizações em Diretórios são tipicamente simples ou nem são feitas.

Diretórios são preparados para dar resposta rápida a um grande volume de consultas ou operações de busca. Eles também podem ter a habilidade de replicar informações extensamente; isto é usado para acrescentar disponibilidade e confiabilidade, enquanto reduzem o tempo de resposta.

Existem várias maneiras diferentes para disponibilizar um serviço de Diretório. Métodos diferentes permitem que diferentes tipos de informações possam ser armazenadas no Diretório, colocando requerimentos diferentes, sobre como aquela informação poderá ser referenciada, requisitada e atualizada, como ela é protegida de acessos não autorizados, etc. Alguns serviços de Diretório são locais, fornecendo o serviço para um contexto restrito (exemplo: o serviço *finger* em uma máquina isolada). Outros serviços são globais, fornecendo o serviço para um contexto muito maior (por exemplo, a própria Internet).

O serviço de Diretório *LDAP* é baseado em um modelo cliente-servidor. Um ou mais servidores *LDAP* contêm os dados criando a árvore de Diretório *LDAP*. Um cliente *LDAP* conecta-se a um servidor e faz uma requisição. O servidor responde com a requisição, ou exibe um ponteiro para um local onde o cliente pode conseguir a informação (tipicamente, outro servidor *LDAP*). Pode-se fazer novamente uma comparação com o *DNS*, a diferença é que o servidor *LDAP* não faz buscas recursivas, ou seja, em nome do cliente. O cliente é encarregado de procurar pelo servidor até encontrar a informação desejada (CONECTIVA, 2004).

Características adicionais

Permite a utilização de soluções baseadas em criptografia com suporte a *SSL (Secure Socket Layer)* e *TLS (Transport Layer Security)*. Tais configurações podem ser ativadas através do arquivo *slapd.conf*.

Conclusões

O serviço de diretórios *LDAP* tem se tornado padrão nos últimos anos, portanto o bom entendimento do funcionamento e da possibilidade de implementá-lo nas mais diversas funcionalidades é atividade crucial e, assim, de responsabilidade do administrador do ambiente de rede.

3.1.5 *PAM*

Definição

PAM (Pluggable Authentication Modules) é um conjunto de bibliotecas compartilhadas proposto pela *SUN Microsystems* e pode ser utilizado em quaisquer distri-

buições *Linux*, além de funcionar no *FreeBSD*. Permitem ao administrador escolher como os usuários serão autenticados nas aplicações.

Tais bibliotecas são chamadas de módulos e têm seus próprios testes, regras e critérios. O administrador do sistema configura arquivos para verificar usuários e programas. Baseado nas respostas desses módulos, a autenticação pode ser efetivada ou rejeitada para determinado usuário a um programa específico.

Funcionamento

Com esse esquema de autenticação o administrador pode escolher qualquer combinação de serviços para prover autenticação. Estão listadas abaixo as principais vantagens do *PAM* para o administrador de sistemas (COMPANY, 2000):

- Política de configuração flexível;
- Política de autenticação por aplicação;
- Possibilidade de escolha do mecanismo de autenticação padrão para aplicações não especificadas;
- Facilidade de uso para usuário final;
- Mapeamento de senhas, que permite uso de senha única, mesmo se a senha é associada a métodos de autenticação diferentes;
- Possibilidade de passar parâmetros opcionais para o serviço.

Os módulos utilizados pelo *PAM* são divididos em quatro tipos básicos de acordo sua função: autenticação, gerenciamento de conta, gerenciamento de sessão e gerenciamento de senha (MORGAN, 2002).

1. Módulos de autenticação: provêem autenticação para seus usuários e permite que as credenciais sejam ativadas, recarregadas ou destruídas. Estes módulos permitem ao usuário ser identificado;
2. Módulos de conta: verificam idade da senha, expiração da conta e restrições de horário de acesso. Uma vez que o usuário foi identificado pelos módulos de autenticação, os módulos de conta determinarão se o usuário pode ter acesso.
3. Módulos de sessão: gerenciam a abertura e fechamento de uma sessão de autenticação;
4. Módulos de senha: permitem alterações de senha e atributos relacionados a ela.

Características adicionais

Dentre os vários modos de autenticação suportados pelo *PAM*, destacam-se o *LDAP*, *Winbind* e *kerberos*.

Conclusões

Esse esquema de autenticação é tão poderoso que a maioria das aplicações que rodam em Linux, por exemplo, o utilizam de alguma forma, ou são, pelo menos, compatíveis.

Portanto, esta é a ferramenta recomendada quando se tem a necessidade de fazer um tratamento no esquema de autenticação para um determinado serviço.

3.1.6 *SARG*

Definição

O *SARG* (Squid Analisys Report Generator) é um analisador de logs do *Squid* capaz de informar ao administrador em um formato bastante agradável por onde os usuários estão navegando na Internet.

Funcionamento

A ferramenta lê os logs do *Squid* por meio de um agendamento que deve ser configurado pelo administrador. Normalmente executado diariamente no mesmo horário.

É gerado um arquivo em formato texto contendo informações úteis sobre a navegação na Internet durante aquele período encontrado no arquivo de log.

Um script *Perl* é executado com o objetivo de mostrar as informações do arquivo *txt* em formato *Web* conforme pode ser visto na Figura 3.3.

Na visualização na *Web*, gráficos dos acessos também podem ser mostrados conforme Figura 3.4.

Características adicionais

- Está disponível em mais de vinte línguas, dentre as quais: português, inglês e espanhol;
- É capaz de ler logs do *ISA Server*, servidor proxy da *Microsoft*;
- É possível customizar os relatórios com informações consideradas relevantes como por exemplo: sítios mais visitados, usuários que visitaram determinados sítios, sítios visitados por determinado usuário, etc (ORSO,).

Squid Analysis Report Generator

Squid User Access Report

Period: 2004Aug06-2004Sep13

Sort: BYTES, reverse

Topuser Report

Topsites Report

Sites & Users Report

Downloads Report

Denied Report

Authentication Failures Report

NUM		USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT		ELAPSED TIME	MILISEC	%TIME
1		user004	16K	918M	23.12%	1.39%	98.61%	08:51:26	32M	1.79%
2		user069	70K	646M	16.26%	4.68%	95.32%	23:00:23	83M	4.66%
3		user264	61K	498M	12.53%	6.14%	93.86%	206:45:02	745M	41.85%
4		user260	68K	387M	9.73%	7.00%	93.00%	139:04:10	501M	28.15%
5		user255	22K	251M	6.31%	3.31%	96.69%	12:07:16	44M	2.45%
6		user180	1K	152M	3.83%	0.33%	99.67%	00:16:15	976K	0.05%
7		user159	469	141M	3.54%	0.05%	99.95%	00:26:10	2M	0.09%
8		user1671	6K	106M	2.66%	7.61%	92.39%	01:00:49	4M	0.21%
9		user241	55K	88M	2.21%	6.68%	93.32%	50:22:18	182M	10.20%
10		user079	1K	80M	2.00%	0.50%	99.50%	00:15:18	918K	0.05%
11		user214	164	58M	1.45%	0.00%	100.00%	00:09:51	591K	0.03%
12		user068	353	48M	1.21%	0.12%	99.88%	00:05:39	340K	0.02%
13		user234	8K	42M	1.05%	4.67%	95.33%	14:10:57	52M	2.87%
14		user023	523	39M	0.97%	2.03%	97.97%	04:26:07	16M	0.90%
15		user082	6K	35M	0.88%	23.25%	76.75%	01:40:47	7M	0.34%
16		user228	6K	33M	0.81%	30.49%	69.51%	00:39:31	3M	0.13%
17		user145	1K	29M	0.71%	51.51%	48.49%	00:07:26	447K	0.03%
18		user224	4K	26M	0.65%	5.33%	94.67%	00:39:50	3M	0.13%
19		user136	4K	19M	0.47%	17.71%	82.29%	01:07:48	5M	0.23%
20		user1678	907	16M	0.39%	5.96%	94.04%	00:13:16	796K	0.04%
21		user265	3K	15M	0.38%	0.83%	99.17%	00:35:00	3M	0.12%
22		user099	91	14M	0.35%	2.23%	97.77%	00:02:40	160K	0.01%
23		user013	3K	12M	0.30%	17.58%	82.42%	00:17:31	2M	0.06%
24		user188	4K	12M	0.28%	16.44%	83.56%	00:54:51	4M	0.19%
25		user242	236	11M	0.27%	2.08%	97.92%	00:02:50	171K	0.01%
26		user187	3K	11M	0.27%	16.77%	83.23%	00:34:10	3M	0.12%
27		user078	6K	11M	0.27%	88.89%	11.11%	00:02:21	141K	0.01%
28		user251	4K	11M	0.26%	7.85%	92.15%	00:21:51	2M	0.07%

Figura 3.3: Estatísticas de acesso no *SARG*.

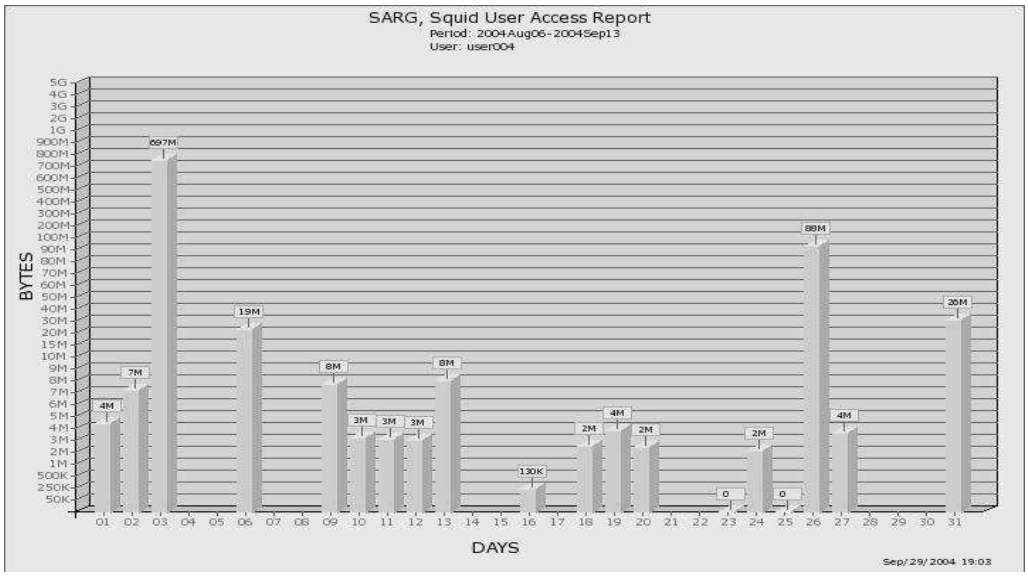

Figura 3.4: Gráficos gerados pelo *SARG*.

Conclusões

O *SARG* é uma ferramenta simples e de fácil implementação e manutenção, que se integra muito bem ao *Squid*.

Os relatórios gerados por esta aplicação são importantes para o corpo gerencial da corporação uma vez que os dados estatísticos produzidos podem servir como embasamento para tomada de decisão

3.2 Validação de Autenticação no Controlador de Domínio

A identificação do usuário quando do acesso a Internet é uma necessidade comum em ambientes corporativos. Muitas empresas consideram importante saber que sítios seus usuários acessam.

Para suprir tal necessidade, algumas soluções podem ser implementadas:

1. **Autenticação de usuário para acesso a Internet.**

   Neste modelo, o usuário é forçado a autenticar-se para obter acesso a Internet.

Em alguns ambientes, duas autenticações são necessárias: uma para ter acesso a rede de computadores da empresa e outra para acesso a Internet. Tal solução pode ser uma boa opção em determinados ambientes em que é importante mostrar ao usuário que o acesso a Internet está sendo monitorado. Esta solução é uma forma de fazê-lo.

2. **Validação de autenticação de usuário para acesso a Internet.**

   Um segunda opção, considerada mais efetiva, é aquela em que a autenticação na rede é suficiente para dar acesso a Internet. Porém, sem causar perdas nos registros que armazenam informações de qual usuário acessou qual sítio.

   A validação de autenticação no controlador de domínio enquadra-se no segundo modelo e é o mais comum, pois é transparente para o usuário e permite controle do acesso por parte dos administradores.

Essas soluções são implementadas com a ajuda de um servidor *Proxy*, responsável pela autenticação ou validação de autenticação do usuário.

Para tanto, faz-se necessário conhecer a funcionalidade do servidor *Proxy* no que tange a autenticação de usuários em controladores de domínio.

O *Squid* é um *Proxy* capaz de autenticar usuários através de três métodos (RFC2617, 1999):

- **Método *basic***;
- **Método *digest***;
- **Método *NTLM***[5].

Tais métodos especificam como o *Squid* recebe o nome de usuário e a senha do cliente e valida esses parâmetros.

Para cada método o *Squid* provê módulos de autenticação também chamados de *helpers* (WESSELS, 2004), conforme Seção 3.2.1.

3.2.1 Métodos de Autenticação

Método *Basic*

Do ponto de vista da segurança, o método de autenticação *basic* é considerado muito fraco, pois nome de usuário e senha trafegam em claro pela rede. Contudo, contempla uma grande variedade de possibilidades, listadas abaixo:

[5]NT Lan Manager

1. *NCSA*

   Armazena nome de usuário e senha em um arquivo texto simples, similar ao arquivo de usuários do *Unix, /etc/passwd*.

2. *LDAP*

   As bibliotecas e os arquivos de configuração do *OpenLDAP* devem ser instalados antes da compilação do *squid_ldap_auth helper*, sendo necessário utilizar no mínimo os argumentos *DN (Distinguished Name)* e o nome do servidor *LDAP*.

3. *MSNT*

   O *MSNT* interfaceia bases de dados para o domínio *Windows NT* através do protocolo *SMB* (Server Message Block).

   Este autenticador permite qualquer usuário validado pelo servidor. Porém, ele também tem a característica de permitir ou bloquear usuários especificados.

4. *Multi-domain-NTLM*

   Similar ao MSNT. Porém, enquanto o MSNT consulta base de dados em até cinco domínios, o Multi-domain-NTLM faz a consulta de acordo com o domínio que o usuário informa ao autenticador.

5. *PAM*

   O *PAM* é a ligação entre os módulos de autenticação (*kerberos, LDAP, smart cards*) e as aplicações que requerem serviços de autenticação (*ssh, ftp, imap, proxy*).

6. *SASL*

   A camada de segurança e autenticação simples (SASL) é um padrão proposto pelo *IETF* (Internet Engineering Task Force) documentado pela *RFC 2222*. Trata-se de um protocolo para negociação de parâmetros de segurança para protocolos orientados a conexão (*FTP, HTTP, SMTP*). Entretanto, o autenticador *SASL* é similar ao autenticador *PAM*. Ele faz a interface com uma biblioteca externa para consultar base de dado de autenticação.

   Pode-se configurar o autenticador para verificar um arquivo de senhas tradicional, um sistema *PAM* ou qualquer outra base de dados suportada.

7. *SMB*

SMB é um outro autenticador para bases de dados *Microsoft*. É um programa escrito em *C*, que executa um *shell script* cada vez que se comunica com o controlador de domínio *Windows*. O *shell script* contém comandos do pacote *Samba*. Assim, é necessário instalar o *Samba* antes de utilizar o autenticador.

8. *YP*

   O autenticador *YP*[6] verifica um diretório *NIS* (Network Information System). Assim, para utilizá-lo é necessário fornecer o nome de domínio *NIS* e o nome da base de dados de senhas.

9. *getpwnam*

   Este autenticador é simplesmente uma interface para a função *getpwnam()* encontrada nas bibliotecas de *C* em sistemas *Unix*. A função *getpwnam()* efetua a verificação no arquivo de senhas para um determinado usuário. Caso utilize-se *NIS* ou *PAM*, as bases de dados a eles associadas são verificadas com o objetivo de autenticar o usuário.

10. *winbind*

    O autenticador *winbind* é um cliente para o *programa winbindd* do *Samba*. Portanto, o *Samba* deve ter sido instalado e o *programa winbindd* deve estar em execução antes de utilizar o autenticador.

Método *Digest*

O método de autenticação *digest* é mais robusto, no que diz respeito a segurança, que o método *basic*. Essa característica ocorre devido ao uso de criptografia para tráfego de nome de usuário e senha.

- ***password***

  Esta é uma implementação de referência para uso de autenticação *Digest* para o *Squid*. Utiliza-se um arquivo de usuários e senhas em texto claro no formato padrão de sistemas *Unix usuário:senha*. O *Squid* não provê quaisquer ferramentas para manter arquivo de senhas neste formato. Portanto, quando tem-se a necessidade de tratar arquivos de senhas nesse formato, esta é uma boa solução.

[6] *YP - Yellow Pages* foi o primeiro nome dado ao serviço *NIS*

Método *NTLM*

Trata-se de um método capaz de comunicar com o protocolo de autenticação de conexão proprietário da *Microsoft*. Também faz uso de criptografia, porém é considerado "pesado"por especialistas devido a forma de comunicação com o controlador de domínio (WESSELS, 2004).

1. ***SMB***

   O autenticador *SMB* para *NTLM* é similar àquele para autenticação *Basic*. Os usuário podem simplesmente entrar com domínio, nome de usuário e senha. Este autenticador pode fazer balanceamento de carga entre múltiplos controladores de domínio.

2. ***winbind***

   Este autenticador é similar ao *winbind* para autenticação *Basic*. Ambos requerem o *programa winbindd* instalado e em execução.

   Dentre os módulos de autenticação suportados pelo *Squid*, alguns são bastante utilizados atualmente devido aos protocolos em funcionamento nos controladores de domínio.

   Em redes, cujo controlador de domínio é o *Windows NT/2000*, é comumente utilizado autenticação através do *winbind*. Algumas soluções baseiam-se no protocolo *SMB*.

   As implementações mais recentes têm utilizado protocolos como *LDAP* ou *Kerberos*, também suportados, nativamente, pelo *Windows 2000*.

   Assim, as seções a seguir tratam de dois dos métodos de autenticação mais utilizados nos dias de hoje: *winbind* e *LDAP*.

3.2.2 Uso do *winbind*

Funcionalidades e benefícios

Winbind é uma solução elegante para o problema de *logon* único em redes compostas por sistemas operacionas diferentes.

Trata-se de um componente do pacote *Samba* em suas versões mais recentes que utiliza uma implementação *Unix* das chamadas *RPC (Remote Procedure Call)* da *Microsoft*, *PAM (Pluggable Authentication Modules)* e o *NSS (Name Service Switch)* para permitir a usuários do domínio *Windows NT* funcionarem como usuários *Unix* em equipamentos *Unix*.(TERPSTRA, 2004)

Assim, o *winbind* provê as seguintes funcionalidades distintas (TERPSTRA, 2004):

- Autenticação das credenciais dos usuários (via *PAM*);
- Resolução de identidade (via *NSS*);
- Manutenção da base de dados *winbind idmap.tdb* que armazena os mapeamentos entre os UIDs/GIDs do *Unix* e SIDs do *Windows*.

Funcionamento

O Winbind trabalha em uma arquitetura cliente/servidor. O programa *winbindd* espera por requisições dos clientes. Essas requisições são geradas pelos clientes *NSS* e *PAM* e são processadas sequencialmente.

Instalação e configuração

- Pré-requisitos
  - *Samba*[7] e *PAM*[8] instalados.
- Configuração do *nsswitch.conf* e das bibliotecas do *winbind*

  A Figura 3.5 mostra a configuração necessária no arquivo *nsswitch.conf* para que o serviço *NSS (Name Service Switch)* funcione juntamente com o *winbind*.

```
passwd: files winbind
shadow: files
group:  files winbind
```

Figura 3.5: Configuração do arquivo *nsswitch.conf*

Contudo, bibliotecas do *winbind* que comunicam com o *NSS* também são necessárias, conforme Figura 3.6.

```
# cp ../samba/source/nsswitch/libnss\_winbind.so /lib
# ln -s /lib/libnss_winbind.so /lib/libnss_winbind.so.2
# /sbin/ldconfig -v | grep winbind
```

Figura 3.6: Comandos de configuração de bibliotecas

[7] http://samba.org/
[8] http://www.kernel.org/pub/linux/libs/pam/

- Configuração do *Samba*

  Configurações do *Samba* devem ser implementadas no arquivo *smb.conf*, inclusive aquelas associadas ao *winbind* como na Figura 3.7.

```
workgroup = MINHAEMPRESA
security = domain
password server = *
encrypt passwords = yes
winbind separator = \\
realm = MINHAEMPRESA
winbind use default
domain = yes
template shell = /bin/bash
template homedir = /home/%D/%U
```

Figura 3.7: Configuração do arquivo *smb.conf* para suporte a *winbind*

Além disso, o servidor *Samba* somente terá acesso a base de dados do domínio se for uma máquina pertencente a este domínio. O cadastro do Servidor *Samba* no domínio é mostrado na Figura 3.8.

```
net join –S PDC –U administrador
```

Figura 3.8: Comando para cadastramento do Servidor *Samba* no domínio

- Inicialização e teste do *winbindd*

  Após a inicialização do *winbind*, é interessante efetuar alguns testes a fim de validar as configurações efetuadas. A Figura 3.9 mostra os comandos que podem ser utilizados para a inicialização, obtenção da lista de usuários e grupos do domínio, obtenção de lista unificada de usuários e grupos locais e do domínio, respectivamente.

3.3 Integração das Ferramentas

Esta Seção trata dos pontos que necessitam de configuração de mais de uma ferramenta para se atingir um objetivo como a validação da autenticação de um usuário ou a geração de relatórios estatísticos de acesso a Internet. São mostrados, inclusive, os conteúdos dos arquivos de configuraçÃo que deve ser modificados para se alcançar aa perfeita integração das ferramentas.

```
# /usr/local/samba/bin/winbindd
# /usr/local/samba/bin/wbinfo -u
# /usr/local/samba/bin/wbinfo -g
# getent passwd
# getent group
```

Figura 3.9: Comandos para inicialização e teste do *winbind*

3.3.1 *Squid* + Autenticação

Uma vez que o servidor *proxy/cache* esteja instalado e configurado, a integração ocorre de forma muito simples. Contudo, a configuração do *Squid* para validação de autenticação no controlador de domínio requer algumas configurações mais apuradas que são o foco desta Seção.

Pré-requisitos

1. *Samba* compilado com suporte *PAM* e a autenticação via *winbind* conforme Figura 3.10.

```
# ./configure --with-automount --with-smbmount \
  --with-pam --with-pam_smbpass \
  --with-acl-support -with-winbind \
  --with-winbind-auth-challenge
# make
# make install
```

Figura 3.10: Compilação do *Samba*

2. *Squid* compilado com suporte ao método de autenticação *NTLM* conforme Figura 3.11.

```
# ./configure --enable-auth=ntlm,basic
# make
# make install
```

Figura 3.11: Compilação do *Squid*

‘

3. *Squid* compilado com suporte a *LDAP* conforme Figura 3.12 para o caso em que autenticação for via *LDAP*.

```
# ./configure --enable-auth-helpers=ldap
# make
# make install
```

Figura 3.12: Compilação do *Squid* com suporte a *LDAP*

4. Modificações no controlador de domínio para permitir o acesso do *Squid*

   Inclusão do arquivo *proxy_auth* no controlador de domínio para permitir que o *Squid* verifique a autenticação dos usuários. O arquivo *proxy_auth* deve ser criado no diretório *netlogon* do controlador de domínio e ter o conteúdo mostrado na Figura 3.13.

```
allow
```

Figura 3.13: Conteúdo do arquivo *proxy_auth*

Configuração do arquivo *squid.conf*

A validação de autenticação efetuada pelo *Squid* no controlador de domínio requer ainda a inclusão dos parâmetros de configuração. São adicionadas linhas associadas a autenticação pelos métodos *NTLM* e *Basic* conforme Figura 3.14.

```
auth_param ntlm program /usr/bin/ntlm_auth \
  --helper-protocol=squid-2.5-ntlmssp
auth_param ntlm children 30
auth_param ntlm max_challenge_reuses 0
auth_param ntlm max_challenge_lifetime 20 minutes

auth_param basic program /usr/bin/ntlm_auth \
  --helper-protocol=squid-2.5-basic
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
```

Figura 3.14: Parâmetros relacionados ao métodos de autenticação

No caso de optar por autenticação usando o protocolo *LDAP*, a configuração deve ser efetuada conforme Figura 3.15.

```
auth_param basic program \
  /usr/local/squid/libexec/squid_ldap_auth \
  -b "ou=empregados,dc=empresa,dc=com" \
  srvldap.empresa.com
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
```

Figura 3.15: Parâmetros relacionados ao métodos de autenticação usando *LDAP*

Além dos parâmetros relacionados ao métodos de autenticação do *Squid* é necessário criar ACLs (Access Control Lists) para liberação de acesso somente aos usuários que forem validados no controlador de domínio conforme Figura 3.16.

```
# Permite acesso somente ao usuarios autenticados
# pertencentes a rede_interna

acl usuarios_autenticados proxy_auth REQUIRED
acl rede_interna src 10.0.0.0/255.0.0.0
http_access allow rede_interna usuarios_autenticados
```

Figura 3.16: *ACLs* para acesso de usuários autenticados

3.3.2 *Squid + Dansguardian*

A união do filtro de conteúdo com o servidor *proxy/cache* traz inúmeras vantagens para aumentar o controle no ambiente de rede quando do acesso a Internet por parte de seus usuários conforme Seção 2.2.

As características do *Squid* e do *Dansguardian* permitem uma integração bastante eficiente.

O *Squid* é um *proxy* extremamente flexível e muito utilizado. Assim, várias ferramentas têm sido criadas com o objetivo de trabalhar em conjunto com o *Squid*, o *Dansguardian* é uma delas.

Configuração do *Squid*

A integração do proxy com o filtro de conteúdo requer o uso de algumas configurações avançadas do *Squid* como utilização das *TAGs cache_peer*, *http_access* e *always_direct*.

A Figura 3.17 mostra o uso de *TAGs*[9] do *Squid* com o objetivo de garantir que as requisições que chegam ao *Squid* sejam encaminhadas ao *Dansguardian* para a filtragem do conteúdo acessado.

```
# Especifica caches numa hierarquia de servidores,
# indicando qual sera o servidor parent, de busca
# para filtragem.
cache_peer srvproxy parent 8080 0 no-query default \
login=*:senha

# Permite acesso do DansGuardian sem autenticacao
http_access allow localhost

# Permite que requisicoes vindas do DansGuardian
# nao sejam reencaminhadas a ele
always_direct allow localhost

# Obriga todas as demais requisicoes a serem
# encaminhadas ao DansGuardian
always_direct deny all
```

Figura 3.17: *TAGs* para filtragem de conteúdo

Configuração do Dansguardian

Na Seção 3.1.2 foram abordadas as características do Dansguardian, das quais é importante ressaltar que não funciona como um *proxy*, mas integrado a ele.

A Figura 3.18 mostra parte do conteúdo do arquivo *dansguardian.conf* em que são informados os caminhos para os arquivos contendo os filtros, o valor total considerado adequado para acesso baseado em pesos e outras customizações.

Os arquivos de filtros podem conter os bloqueios efetivos ou apontamentos para arquivos que contenham listas mais completas. Tais listas, contendo milhares de filtros, encontram-se disponíveis na Internet (BARRON, 2003a).

Os principais filtros são realizados da seguinte forma:

site Lista de domínios bloqueados conforme Figura 3.19;

url Lista de *URLs* bloqueados conforme Figura 3.20;

[9] São seções dentro do arquivo de configuração do *Squid*

```
# Localizacao dos arquivos com filtros de conteudo
bannedphraselist = '/etc/dansguardian/bannedphraselist'
bannedurllist = '/etc/dansguardian/bannedurllist'
bannedsitelist = '/etc/dansguardian/bannedsitelist'

# Valor Peso
naughtynesslimit = 100
```

Figura 3.18: Arquivo de configuração do *Dansguardian*

```
# Lista de sites bloqueados

badboys.com
```

Figura 3.19: Domínios bloqueados

```
# Lista de URLs bloqueadas

members.home.net/uporn
```

Figura 3.20: *URLs* bloqueadas

phrase Lista de frases ou termos bloqueados conforme Figura 3.21;

```
# Bloqueio da frase "sex magazine".
 <sex magazine>

# Bloqueio de ápginas contendo "sex" e "fetish".
 <sex>,<fetish>
```

Figura 3.21: Frases ou termos bloqueados

weightedphrase Lista de frase ou termos bloqueados segundo pesos, conforme Figura 3.22;

ip Lista de endereços *IP* bloqueados conforme Figura 3.23;

extension Lista de extensões de arquivos bloqueados conforme Figura 3.24;

mimetype Lista de tipos de arquivos bloqueados conforme Figura 3.25;

```
# Lista de frases com pesos associados

<game files><50>
<free game demos><50>
<game files><50>
```

Figura 3.22: Frases ou termos bloqueados por peso

```
# Lista de IPs bloqueados

192.168.0.1
192.168.0.2
```

Figura 3.23: Endereços *IP* bloqueados

```
# Lista de extensoes bloqueadas

.asx  # Windows Media Audio / Video
.bas  # Microsoft Visual Basic class module
.bat  # Batch file
```

Figura 3.24: Extensões de arquivos bloqueados

```
# Lista de MIME types bloqueados

audio/mpeg
audio/x-mpeg
video/mpeg
```

Figura 3.25: Tipos de arquivos bloqueados

3.3.3 *Squid + SARG*

A utilização de um servidor *proxy/cache* como concentrador das requisições provenientes de clientes da rede interna da organização traz consigo a necessidade da geração de relatórios de acesso a Internet, importantes para a administração da empresa.

Esses relatórios podem ser gerados através de ferramentas capazes de ler os arquivos de registro (*logs*) do *Squid* que disponibilizem tais relatórios automatica-

mente e em formato de fácil leitura.

Também é um pré-requisito dessas ferramentas que os relatórios gerados sejam capazes de identificar usuários/máquinas e os sítios acessados.

Todas as necessidades citadas são providas pelo *SARG* quando integrado ao *Squid*, escopo desta Seção.

Configuração do *Squid*

A única prerrogativa necessária no *Squid* é que os registros (*logs*) estejam sendo gerados. Para tanto, a *TAGs cache_access_log* deve ter a configuração mostrada na Figura 3.26

```
# Logs de requisicoes dos clientes
cache_access_log /var/log/squid/access.log
```

Figura 3.26: Tipos de arquivos bloqueados

Configuração do *SARG*

O *SARG* busca o arquivo de registro (*log*) de acesso do *Squid* no diretório padrão de instalação do *Squid*, portanto é importante confirmar que a localização do arquivo de registro procurado pelo *SARG* realmente esteja no lugar de procura.

As configurações adicionais mais importantes a serem realizadas no arquivo *sarg.conf* estão descritas abaixo (CISNEIROS, 2003):

language Idioma utilizado conforme Figura 3.27;

```
# Lingua selecionada
language Portuguese
```

Figura 3.27: Seleção de idioma

access_log Arquivo de registro (*log*) do *Squid* conforme Figura 3.28;

```
# Arquivo de LOG do Squid que alimentara o SARG
access_log /var/log/squid/access.log
```

Figura 3.28: Seleção de arquivo de registro (*log*)

```
# Titulo da pagina principal
title "Relatorio do Squid Proxy"
```

Figura 3.29: Seleção de Título de Relatório

title Título da página de relatório conforme Figura 3.29;

output_dir Diretório de saída das páginas de relatório conforme Figura 3.30;

```
# Diretorio de Saida para as paginas de relatorio
output_dir /var/www/html/squid-reports
```

Figura 3.30: Seleção de diretório para geração de relatório

exclude_users Usuários que não devem estar presentes no relatório conforme Figura 3.31;

```
# Usuarios que nao devem estar nos relatorios
exclude_users /etc/sarg/exclude.users
```

Figura 3.31: Seleção de usuários excluídos

exclude_hosts Máquinas que não devem estar presentes no relatório conforme Figura 3.32;

```
# Maquinas que nao devem estar nos relatorios
exclude_hosts /etc/sarg/exclude.hosts
```

Figura 3.32: Seleção de máquinas excluídas

date_format Formato da data conforme Figura 3.33.

```
# Formato da data usada no Brasil dd/mm/aa
date_format e
```

Figura 3.33: Seleção de formato da data

3.3.4 Inicialização e Reinicialização das Ferramentas

Importante observar que as ferramentas e os serviços abordados neste Capítulo devem ser inicializados ou recarregados de acordo com as indicações da documentação de cada um deles.

Neste trabalho, foi utilizado o sistema operacional *Fedora Linux Core 3* e as instalações de programas e serviços neste ambiente recomendam a utilização do comando *service* conforme Figura 3.34.

```
# Squid inicia, reinicia e recarrega
service squid start
service squid restart
service squid reload

# Dansguardian inicia, reinicia e recarrega
service dansguardian start
service dansguardian restart
service dansuardian reload

# SARG Geracao de Relatorio
/usr/sbin/sarg -f /etc/sarg/sarg.conf
```

Figura 3.34: Comandos de ativação de serviços

Capítulo 4

Conclusões

O acesso a Internet nas empresas é fato nos dias de hoje. A intenção de disseminar o uso quando as primeiras empresas passaram a utilizá-la levou o público corporativo a níveis de aceitação tão altos que transformou-se em preocupação por parte da gerência das instituições. Contudo, uma vez o serviço havia sido disponibilizado e bem aceito, gerou a dificuldade em retirá-lo.

Sabe-se que uma política de segurança adotada na instituição poderia limitar, ou restringir completamente os acessos. Contudo, existe uma dificuldade ainda maior nas empresas de implementar tal política, que requer apoio da alta cúpula da instituição.

Nesse sentido, os setores responsáveis pela Informática nas empresas passaram a buscar soluções alternativas capazes de permitir o acesso a rede mundial de computadores com restrições reconhecidamente necessárias.

A utilização de ferramentas baseadas em Software Livre, licenciadas sob *GPL* mostrou-se possível, econômica e eficiente em ambientes corporativos.

Estudos aprofundados de ferramentas e soluções existentes permitiram resolver problemas complexos sem custos com licenciamento de programas.

Este trabalho resultou de tais estudos, passando pela seleção das ferramentas até a implementação em ambientes corporativos. Portanto, espera-se que este trabalho sirva como base para iniciativas nesse sentido.

A questão de limitação de acessos a Internet do ponto de vista de privacidade dos funcionários das empresas que são monitorados pela solução não foi analisada, uma vez que o escopo do trabalho foi estritamente técnico.

Esta implementação de controle de acesso a Internet em ambientes corporativos contempla o acesso *Web*, efetuado através dos protocolos *HTTP*, *HTTPS* e *FTP*. Entretanto, outros tipos de acessos são realizados constantemente nas empresas: uso de clientes de correio eletrônico, clientes de acesso remoto, ferramentas

de *scanner* de portas e tentativa de acesso a portas de serviços.

Como trabalho futuro, tem-se o objetivo de acrescentar a solução implementada capacidade de monitorar outros serviços como:

Correio Eletrônico Protocolos *POP3*, *IMAP* e *SMTP*;[1].

Acesso Remoto Clientes *Telnet*, *SSH*[2], *VNC* e outros;

Verificação de atividade de serviços *Scanners* de vulnerabilidades e portas, *pings* e *traceroutes*.

Para tanto, a intenção é adicionar a solução a implementação de *Firewall* integrado a ferramenta de detecção de intrusão, utilizando ferramentas como o *iptables*[3] e *snort*[4].

[1] Protocolos da camada de aplicação da pilha de protocolos *TCPIP*
[2] Protocolos da camada de aplicação da pilha de protocolos *TCPIP*
[3] http://www.netfilter.org
[4] http://www.snort.org

Referências Bibliográficas

BARRON, D. *Blacklists*. 2003. Http://dansguardian.org/?page=blacklist. Visitado em março de 2005.

BARRON, D. *Dansguardian Introduction*. 2003. Http://dansguardian.org/?page=introduction. Visitado em janeiro de 2005.

CISNEIROS, H. *Gerando relatórios do Squid com o SARG*. 2003. Http://www.devin.com.br/eitch/sarg/. Visitado em março de 2005.

COMPANY, S. M. C. *PAM Administration*. 2000. Http://www.sun.com/software/solaris/pam/pam.admin.pdf. Visitado em fevereiro de 2005.

CONECTIVA, E. *Guia do Servidor Conectiva Linux 10*. [S.l.], 2004. Http://www.conectiva.com/doc/livros/online/10.0/servidor/pt_BR/ch13s02.html. Visitado em janeiro de 2005.

FOUNDATION, L. *OpenLDAP 2.0 Administrator's Guide*. 2004. Http://www.openldap.org/doc/index.html. Visitado em janeiro de 2005.

MORGAN, A. G. *Linux-PAM System Administrators Guide*. 2002. Http://www.kernel.org/pub/linux/libs/pam/index.html. Visitado em janeiro de 2005.

ORSO, P. *Squid Analysis Report Generator*. Http://sarg.sourceforge.net/. Visitado em março de 2005.

RFC2617. 1999. Http://www.ietf.org/rfc/rfc2617.txt?number=2617. Visitado em março de 2005.

SOFTWARE, S. B. *Internet Filtering Alternatives White Paper*. 2003. Http://www.stbernard.com/products/docs/Internet_Filtering_Alternatives.pdf. Visitado em fevereiro de 2005.

SZTOLTZ, R. S. T. e. E. d. O. F. R. L. *Guia do Servidor Conectiva 9*. [S.l.], 2003. Http://www.conectiva.com/doc/livros/online/9.0/servidor/autenticacao.html. Visitado em janeiro de 2005.

TERPSTRA, J. R. V. J. H. *The Official Samba-3 HOWTO and Reference Guide*. 2nd. ed. [S.l.]: Prentice Hall PTR, 2004.

WESSELS, D. *Squid: The Definitive Guide*. [S.l.]: O'Reilly Media, Inc., 2004.