Controle de acesso através do Squid

EDUARDO AUGUSTO COSA[1]

[1]Curso ARL - DCC / UFLA - Cx Postal 37 - CEP 37200-000 Lavras (MG)
eacosa@gmail.com

Resumo: *Este artigo apresenta a ferramenta de* proxy/cache *Squid como uma alternativa para a implementação de um controle de acesso à internet, objetivando desta forma coibir o uso inadequado, principalmente em ambientes corporativos.*

Palavras-Chave: *Squid, controle de acesso,* proxy, cache.

1 Introdução

Com o aumento dos incidentes de segurança da informação e o rápido crescimento da internet no meio corporativo, a preocupação com segurança da informação é cada vez maior. A política de segurança é uma peça chave quando queremos tornar um ambiente computacional mais seguro. Uma Política de Segurança é um conjunto de leis, regras e práticas que regulam como uma organização gerencia, protege e distribui suas informações e recursos (UCHÔA, 2003).

Umas das preocupações que deve ser abordada na política de segurança diz respeito ao controle de acesso, ou seja, o que o usuário pode ou não estar fazendo. Os mecanismos para este controle não devem ser abordados pela política de segurança, mas devem ser implementados de maneira que se faça cumprir o que nela é determinado. Mais detalhes sobre política de segurança podem ser encontrados em (NIC BR SECURITY OFFICE, 2003).

Em se tratando de acesso a internet existem várias formas de realizar o controle, a forma mais comum é o controle através de *firewalls* baseados em filtro de pacotes e sistemas de *proxy*. O *proxy* é um programa que fica entre a rede local e a rede pública (internet), realizando o controle na comunicação entre os dois lados (UCHÔA; SIMEONE; SICA, 2003).

O *proxy* trabalha como um intermediário entre cliente e o servidor, ou seja, ele recebe as requisições e repassa aos servidores. Essa característica pode gerar uma confusão com o NAT – *Network Address Translation* (SRISURESH; EGEVANG, 2001), porém o *proxy*, diferente do NAT, trabalha baseado na aplicação. Por trabalhar com uma aplicação específica, o *proxy* permite um controle maior sobre várias aplicações, como as que podem usar qualquer porta, uma vez que ele trabalha em portas e aplicações pré-definidas (RUFINO, 2002).

O Squid é um poderoso servidor de *proxy/cache* que suporta os protocolos FTP (POSTEL; REYNOLDS, 1985) e HTTP (FIELDING *et al.*, 1999), oferecendo 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 GPL - GNU *General Public License* (FREE SOFTWARE FOUNDATION, 1991), o Squid é largamente utilizado para compartilhamento de acesso a WEB, possuindo características que permitem ainda trabalhar com outros objetivos como melhoria de performance e controle de acesso.

Nesse contexto, o objetivo deste artigo é abordar as principais configurações para a implementação de um controle de acesso através do uso do *proxy/cache* Squid. Para isso, o texto encontra-se organizado da seguinte forma: nas Seção 2 e a Secão 3 são apresentados, respectivamente, os parâmetros e os mecanismos de controle de acesso no Squid; os mecanismos de autenticação de usuários no Squid são mostrados na Seção 4; na Seção 5, são avaliadas duas ferramentas para análise de *logs* do Squid, SARG e Calamaris. Por fim, na Seção 6 é apresentado resultados obtidos com o uso de técnicas apresentadas neste trabalho.

2 Alguns parâmetros de controle de acesso

O controle de acesso no Squid é configurado via arquivo de configuração `squid.conf`, através de alguns parâmetros, apresentados na Tabela 1. Esses parâmetros, em geral, estão associadas a uma dada ACL (*Access Control List* – Lista de Controle de Acesso), que definem o contexto de um determinado controle de acesso.

Como visto na Tabela 1, o parâmetro acl é utilizado para definir uma dada ACL. Existem vários tipos de ACLs, os mais importantes são listados na Tabela 2. Sua sintaxe possui a forma: "`acl nome_acl tipo_da_acl informacao`". Detathes podem ser verificados em (VISOLVE.COM, 2002).

As listas de controle de acesso (ACLs) são interpretadas pelo Squid de cima para baixo, portanto deve-se ter o cuidado no momento de estabelecer à ordem das regras. Ela é muito importante, pois encontrada uma regra que venha a coincidir com determinada ação, as demais regras não serão checadas.

Alguns tipos são pouco utilizados, como *srcdomain*, *urlpath_regex*, *port*, *proto*, *method*, *browser*, *ident* e *arp*, não sendo listados na Tabela 2. Além desses, existem variantes de alguns que permitem a utilização de expressões regulares, como *srcdomain_regex*, *dstdomain_regex*, *ident_regex* e *proxy_auth_regex*.

3 Mecanismos de controle de acesso

O objetivo desta seção é apresentar alguns mecanismos de controle de acesso, ou seja, algumas implementações, utilizando os parâmetros vistos na Seção 2.

3.1 Controle de Acesso por Endereçamento IP

```
acl LOCAL_NET src 192.168.10.0/24
http_access allow LOCAL_NET
http_access deny all
```

Esta regra é bastante simples, mas faz parte de praticamente toda configuração segura do Squid. Ela garante o acesso da rede local (192.168.10.0/24). Na primeira linha, foi

Tabela 1: Parâmetros do Squid

Parâmetro	Descrição
icp_access	ICP – Internet Cache Protocol (WESSELS; CLAFFY, 1997) é um protocolo UDP que tem como objetivo permitir o compartilhamento de informações entre servidores *cache*. Quando se tem uma hierarquia de servidores *cache* configurados, uma das possibilidades de comunicação é através do protocolo ICP. O *icp_access* é responsável por liberar ou não o acesso ICP a uma determinada ACL.
miss_access	Assim como o *icp_access*, o *miss_access* é usado quando se trabalha dentro de uma hierarquia de servidores *cache*. Ele determina como será atendida a solicitação por um "vizinho", de um objeto que não esteja armazenado localmente. Em (FONSECA, 1998) são apresentados mais detalhes e conceitos sobre hierarquias de servidores *cache*.
cache_peer_access	Este parâmetro é utilizado para limitar ou mesmo direcionar uma determinada ACL a um determinado servidor *cache*. Pode ser utilizado, por exemplo, quando se deseja que uma determinada rede utilize um servidor de cache específico.
proxy_auth_realm	Quando se utiliza autenticação no Squid, uma janela solicitando nome de usuário e senha será apresentada para o usuário. Nessa janela é apresentado a identificação do servidor, configurada no parâmetro *proxy_auth_realm* (JUCÁ, 2003).
http_access	O parâmetro *http_access* é responsável por liberar ou não o acesso HTTP a uma determinada ACL.
acl	O parâmetro *acl* é o elemento principal das ACLs, sendo responsável pela sua implementação.

dada o nome de `LOCAL_NET` à ACL e associada a ela todas requisições com origem na classe IP da rede local, utilizando o tipo *src*. Na segunda linha, foi liberado o acesso às requisições que coincidam com as características da ACL `LOCAL_NET`; e, na terceira linha, negou-se o acesso a outras máquinas.

3.2 Controle de acesso pelo nome de domínio do destino

```
acl SITES_PROIB dstdomain   www.sexy.com.br  .playboy.com.br
http_access allow !SITES_PROIB
http_access deny all
```

Neste exemplo, tem-se três recursos importantes sendo utilizados. O "." (ponto) antes da indicação de um endereço indica nome de domínio, incluindo todos os seus servidores. O sinal de "!" (exclamação) funciona como uma negação. No caso, será permitido o acesso a qualquer servidor, com exceção daqueles listados no parâmetro `acl`. Observe

Tabela 2: Principais Tipos de ACLs no Squid

Tipo	Descrição
src	Utilizado para indicar endereços IP de origem (IP do cliente), podendo indicar o endereço de um *host*, uma faixa ou mesmo uma classe de endereços IP.
dst	Indica endereços IP de destino, também pode trabalhar com o endereço de um *host*, uma faixa ou uma classe de endereços IP. Antes de interpretar uma ACL deste tipo, o Squid faz uma consulta DNS para a identificação do IP do endereço que vai no cabeçalho da requisição.
dstdomain	Situação semelhante à apresentada anteriormente, indica o domínio de destino. Neste caso, não existe pesquisa reversa ao servidor DNS para a identificação do IP do cliente, por estar tratando a regra do domínio de destino da requisição.
time	Este tipo permite que seja configurado regras de acordo com o dia da semana e o horário de acesso. Os dias da semana são indicados por meio de iniciais do dia da semana em língua inglesa. A indicação do horário deve ser feita através de um intervalo. Sua sintaxe é na forma: "`acl NOME time [dias_da_semana] [hh1:mm1-hh2:mm2]`"
url_regex	Pesquisa na URL pela expressão regular indicada. Este tipo é *case sensitive*, ou seja faz a diferenciação entre *strings* de caixa alta e caixa baixa. Caso não seja de interesse esta característica, deve-se usá-lo com a opção `-i`.
proxy_auth	Este parâmetro faz com que o Squid entenda que deve trabalhar com autenticação de usuário através de um sistema de autenticação externo.
snmp_community	Este tipo é utilizado para controlar o acesso ao Squid através do protocolo de gerenciamento SNMP (CASE *et al.*, 1990).
maxconn	Este tipo permite controlar o número máximo de conexões de um determinado cliente. Para que seja possível o uso deste tipo, deve-se ter a parâmetro *client_db* ativo no arquivo de configuração do Squid, por padrão, encontra-se habilitado.
req_mime_type	Mais um tipo que faz uso da expressão regular, neste caso para identificar a *string* informada dentro do tipo MIME do cabeçalho da requisição.

ainda que foi informada uma lista de itens para o tipo *dstdomain*. Alguns administradores, inclusive, preferem criar essa lista em arquivo à parte, configurando a chamada de forma similar à:

```
acl SITES_PROIB dstdomain  ''path/para/arquivo''
```

Nesse caso, "`path/para/arquivo`" é o caminho local do arquivo com a relação de endereços. Esse recurso pode ser utilizado em praticamente todos os tipos de ACLs no Squid.

3.3 Controle de acesso pelo dia/hora da semana com máximo de conexão

```
acl EXEMPLO_T time MTWHF 11:00-13:00
acl EXEMPLO_M maxconn 4
http_access allow EXEMPLO_T EXEMPLO_M
http_access deny all
```

Neste exemplo, está sendo liberando acesso integral, com número máximo de quatro conexões simultâneas, de Segunda a Sexta-Feira no intervalo que vai das 11 horas às 13 horas.

3.4 Controle de acesso baseado em palavras chave

```
acl PROIBIDO url_regex -i ''$SQUID-HOME/etc/CONT_PROIBIDO''
http_access allow !PROIBIDO
acl QUERY url_regex cgi{}-bin ?
no_cache deny QUERY
http_access deny all
```

Neste tipo de controle, foi feito uso de um arquivo externo, no diretório `etc/` do diretório *home* do Squid, denominado `CONT_PROIBIDO`. Nele, serão colocadas as palavras que quando encontradas deverão barrar o acesso ao site. Além disso, foi informado ao Squid para não fazer cache de páginas geradas dinamicamente via CGI.

Observe que, em uma configuração real, as ACLs são, normalmente, declaradas no início, com regras em seguida. Neste exemplo, optou-se por uma forma alternativa de apresentação (também aceita pelo Squid), com finalidades didáticas.

3.5 Controle por MIME

```
acl EXEMPLO req_mime_type ^application/x-msn-messenger$
http_access deny EXEMPLO
```

Muitos programas que utilizam a tecnologia P2P, como kazaa, MSN entre outros utilizam a porta 80 para realizarem a comunicação, o que acaba dificultando seu bloqueio através do firewall. Uma forma de barrar este acesso é através do tipo MIME do cabeçalho, como mostrado no exemplo anterior, ele objetiva barrar a utilização da porta 80 pelo MSN. Nesse exemplo, foram utilizadas expressões regulares, para maiores detalhes sobre o uso desse tipo de recurso, ver (JARGAS, 2001).

4 Autenticação no Squid

O Squid permite que seja realizado um controle de acesso baseado em usuários, ou seja, os usuários para conseguirem o acesso a internet devem, preliminarmente, se autenticar no servidor *proxy*. Essa autenticação pode ser realizada de diversas maneiras, sendo as mais comuns o formato NCSA (o mesmo utilizado no servidor WEB apache), através de um PDC Windows NT/2000/2003, através de um servidor LDAP, através de módulos PAM, entre outros.

Por padrão, o Squid não traz configuração de autenticação habilitada, portanto devem ser realizados alguns ajustes no arquivo `squid.conf`, mais especificamente nas sessões referentes a: parâmetros de autenticação (*auth_param*) e Lista de Controle (ACL). Para que o Squid passe a solicitar a autenticação do usuário, duas linhas devem ser acrescentadas na lista de controles, são elas:

```
acl name proxy_auth REQUIRED
http_access allow name
```

É necessário agora informar ao Squid a configuração de autenticação que a ACL acima deverá utilizar e para isso deve-se configurar a diretiva *auth_param*, que irá especificar o tipo de autenticação utilizada. Observe que os módulos de autenticação devem ser compilados a parte, sendo encontrados no diretório `$SQUID-SRC/helpers/basic_auth`. O processo de instalação desse módulo é extremamente simples, com boa documentação. O restante da seção apresenta detalhes de algumas formas de autenticação.

Para uma implementação simples de autenticação, utilizando-se o formato NCSA, as seguintes linhas devem ser implementadas no arquivo de configuração:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Proxy Server
auth_param basic credentialsttl 4 hours
```

Na primeira linha, foi indicado o caminho do módulo de autenticação que será utilizado e onde será criado o arquivo de usuários para autenticação. Na segunda linha, configurou-se quantos processos filhos do módulo de autenticação poderão existir, número que deve variar bastante de acordo com o tamanho da rede. Na terceira linha, é informado o título da janela que irá solicitar a senha ao usuário e, por fim, na quarta, é indicado o tempo de vida de uma autenticação bem sucedida. Para criação e inserção de usuários deve ser utilizado o comando `htpasswd`. Esse comando é o mesmo utilizado na implementação de autenticação nos servidores apache.

Para utilizar a autenticação em servidores PDC (Windows ou SAMBA), é necessário atentar que, após a compilação, dois arquivos serão criados no diretório `etc/` do Squid: `msntauth.conf` e `msntauth.conf.default`. O primeiro deve ser editado de acordo com a configuração local. Segue-se um exemplo desse arquivo:

```
# MSNT authenticator configuration file
```

```
# NT hosts to use. Best to put their IP
# addresses in /etc/hosts.
server pegasus pegasus eacosa.com
#denyusers /usr/local/squid/etc/msntauth.denyusers
#allowusers /usr/local/squid/etc/msntauth.allowusers
```

Nesse arquivo, é indicado o PDC do domínio, bem como o BDC (caso exista). Caso se pretenda, é possível ainda utilizar dois arquivos para controlar os usuários que têm ou não permissão de acesso a este serviço de autenticação. Em nosso ambiente, testes foram realizados utilizando-se uma rede com apenas um PDC, por isso repetimos o nome do servidor na entrada referente ao BDC e no caso não foram utilizados os arquivos de controle de usuiáros. Outro detalhe importante é que deve-se adicionar no arquivo `/etc/hosts` do servidor Squid o endereço IP do PDC. Com relação a lista de controles, a implementação é similar à usada na autenticação NCSA:

```
auth_param basic program /usr/lib/squid/msnt_auth
auth_param basic children 5
auth_param basic realm Squid Proxy Server
auth_param basic credentialsttl 4 hours
```

Para utilizar PAM para a autenticação dos usuários, após a compilação do módulo, é necessária a configuração do arquivo squid dentro de `/etc/pam.d`, com as seguintes linhas:

```
auth required /lib/security/pam_pwdb.so shadow nullok
account required /lib/security/pam_pwdb.so
```

No arquivo `squid.conf`, a implementação é semelhante às implementações exemplificadas anteriormente, com uma pequena diferença na primeira linha, onde é necessário indicar o arquivos de usuários e senhas, no caso `/etc/shadow`.

```
auth_param basic program /usr/lib/squid/pam_auth  /etc/shadow
auth_param basic children 5
auth_param basic realm Squid Proxy Server
auth_param basic credentialsttl 4 hours
```

5 Ferramentas de Análise de Logs do Squid

Tão importante como controlar o acesso é acompanhar e interpretar os logs gerados pelo Squid. Com uso de ferramentas auxilares, é possível analisar uma instalação e os resultados obtidos, possibilitando acompanhamento e refinamento da configuração. Em nosso conhecimento, destacam-se duas ferramentas: Calamaris e SARG.

O Calamaris é uma ferramenta desenvolvida em Perl, sob licença GPL, de uso bastante simples. Esta ferramenta permite a geração de relatórios estatísticos ricos em detalhes em diferentes formatos, entre eles HTML e TXT e a criação de relatórios não apenas para o

Squid, mas também para outras ferramentas, entre elas: Netcache, Oops! Proxy Server, Novell Internet Caching System, entre outros. Sendo desenvolvido em Perl, não existe a necessidade de compilação para o uso. O *download* do arquivo, pode ser feito diretamente do *site*[1]. O calamaris pode ser executado de duas maneiras:

```
# cat /var/log/squid/access.log | /usr/bin/calamaris -a
```

ou

```
# /usr/bin/calamaris -a -F html /var/log/squid/access.log \
> /path/do/destino/
```

No primeiro caso, o Calamaris irá gerar todos os relatórios possíveis e imprimi-los na tela do terminal. No segundo exemplo, é solicitado que ele gere todas as estatísticas (parâmetro `-a`) no formato html (parâmetro `-F html`), no diretório indicado. É possível consultar exemplos de relatórios em formato TXT no endereço `http://cord.de/tools/squid/calamaris/calamaris-2.out`.

O SARG – Squid Analysis Report Generator é uma ferramenta desenvolvida em C, por um brasileiro, que permite acompanhar através de relatórios os sites acessados pelo seus usuários.

Os relatórios gerados pelo SARG são de simples compreensão e bem completos no que se propõe, além do usuário e do *site* acessado ele apresenta ainda informações como total de conexões, *bytes* tráfegados, se o acesso a determinado site foi negado, data e horário de acesso e etc. Atualmente em sua versão 2.0.4, tem opção para mais de 18 idiomas, entre eles o Português, sendo parte integrante das principais distribuições Linux e considerado hoje uma das principais ferramentas de análise de *logs* do Squid.

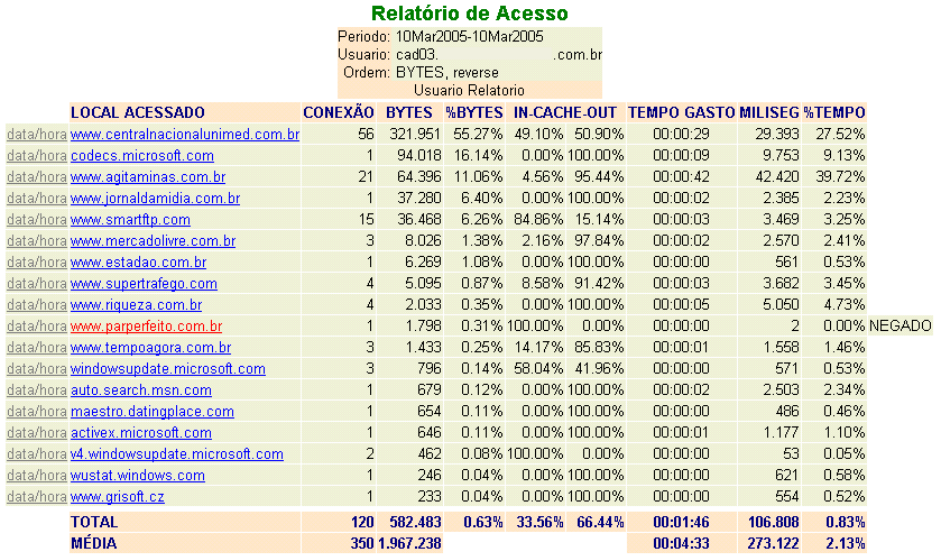
O *download* do SARG pode ser feito diretamente no *site* [2]. Sua configuração também é simples, o único arquivo de configuração é bem documentado, facilitando a configuração. A Figura 1 apresenta um exemplo de relatório gerado pelo SARG.

6 Análise de um problema local

Com a contratação de um *link* dedicado de 256 Kbps para acesso a internet e a liberação do acesso a WEB para todos os computadores da rede local, incluindo mais de 50 computadores acessando deliberadamente a internet, foi identificado que a utilização da WEB não estava sendo feita dentro dos propósitos almejados pela empresa. A solução identificada para implementar o controle no acesso e desta forma melhorar o uso dos recursos disponíveis, evitando o uso inadequado, indiscriminado foi a implementação de um servidor *proxy*.

A escolha pelo Squid como solução ocorreu devido às seguintes características: licença GPL, documentação satisfatória na internet, facilidade no intercâmbio de informações com outros usuários através de listas de discussão, grande flexibilidade no controle de acesso a WEB. O Squid deveria atender ao seguinte escopo em suas ACLs:

[1] Calamaris: `http://cord.de/tools/squid/calamaris/Welcome.html.en`
[2] SARG: `http://sarg.sourceforge.net/`



Relatório de Acesso

Periodo: 10Mar2005-10Mar2005
Usuario: cad03. .com.br
Ordem: BYTES, reverse
Usuario Relatorio

	LOCAL ACESSADO	CONEXÃO	BYTES	%BYTES	IN-CACHE-OUT		TEMPO GASTO	MILISEG	%TEMPO	
data/hora	www.centralnacionalunimed.com.br	56	321.951	55.27%	49.10%	50.90%	00:00:29	29.393	27.52%	
data/hora	codecs.microsoft.com	1	94.018	16.14%	0.00%	100.00%	00:00:09	9.753	9.13%	
data/hora	www.agitaminas.com.br	21	64.396	11.06%	4.56%	95.44%	00:00:42	42.420	39.72%	
data/hora	www.jornaldamidia.com.br	1	37.280	6.40%	0.00%	100.00%	00:00:02	2.385	2.23%	
data/hora	www.smartftp.com	15	36.468	6.26%	84.86%	15.14%	00:00:03	3.469	3.25%	
data/hora	www.mercadolivre.com.br	3	8.026	1.38%	2.16%	97.84%	00:00:02	2.570	2.41%	
data/hora	www.estadao.com.br	1	6.269	1.08%	0.00%	100.00%	00:00:00	561	0.53%	
data/hora	www.supertrafego.com	4	5.095	0.87%	8.58%	91.42%	00:00:03	3.682	3.45%	
data/hora	www.riqueza.com.br	4	2.033	0.35%	0.00%	100.00%	00:00:05	5.050	4.73%	
data/hora	www.parperfeito.com.br	1	1.798	0.31%	100.00%	0.00%	00:00:00	2	0.00%	NEGADO
data/hora	www.tempoagora.com.br	3	1.433	0.25%	14.17%	85.83%	00:00:01	1.558	1.46%	
data/hora	windowsupdate.microsoft.com	3	796	0.14%	58.04%	41.96%	00:00:00	571	0.53%	
data/hora	auto.search.msn.com	1	679	0.12%	0.00%	100.00%	00:00:02	2.503	2.34%	
data/hora	maestro.datingplace.com	1	654	0.11%	0.00%	100.00%	00:00:00	486	0.46%	
data/hora	activex.microsoft.com	1	646	0.11%	0.00%	100.00%	00:00:01	1.177	1.10%	
data/hora	v4.windowsupdate.microsoft.com	2	462	0.08%	100.00%	0.00%	00:00:00	53	0.05%	
data/hora	wustat.windows.com	1	246	0.04%	0.00%	100.00%	00:00:00	621	0.58%	
data/hora	www.grisoft.cz	1	233	0.04%	0.00%	100.00%	00:00:00	554	0.52%	
	TOTAL	**120**	**582.483**	**0.63%**	**33.56%**	**66.44%**	**00:01:46**	**106.808**	**0.83%**	
	MÉDIA	**350**	**1.967.238**				**00:04:33**	**273.122**	**2.13%**	

Figura 1: Relatório de acesso gerado pelo SARG

1. Acesso totalmente liberado para a diretoria, sem restrição de horários e sites;
2. Acesso para gerência e supervisores, restrito ao horário de trabalho (Segunda a Sexta-feira das 07:00-19:00), com restrição a alguns sites;
3. Acesso aos demais colaboradores restrito a sites de trabalho e restrito ao horário de trabalho;
4. Relatórios diários de acesso de todas as máquinas.

As linhas de configuração adicionadas na seção adequada do arquivo `squid.conf` para atender essas necessidades são apresentadas na Figura 2.

```
acl all src 0.0.0.0/0.0.0.0
acl diretoria src 192.168.10.52-192.168.10.53/32
acl cargoconfianca src 192.168.10.54-192.168.10.64/32
acl sitesbloqueados url_regex ''/etc/sitesbloqueados''
acl sitesdetrabalho url_regex ''/etc/sitesliberados''
acl horariotrabalho time MTWHF 07:00-19:00
http_access allow diretoria
http_access allow cargoconfianca horariotrabalho !sitesbloqueados
http_access allow all horariotrabalho sitesdetrabalho
http_access deny all
```

Figura 2: Linhas adicionadas ao `squid.conf` para o controle de acesso

A estratégia de implementação definida junto à diretoria da empresa foi a seguinte: em um primeiro momento foi implementada apenas a geração dos relatórios diários, onde foi acompanhado o acesso dos usuários durante uma semana, relacionando os sites que estavam sendo acessados e que deveriam ser bloqueados. Na semana seguinte foi implementado o bloqueio dos sites relacionados na semana anterior, acrescidos de uma relação de sites disponível na internet[3], criando uma lista negra de *sites* pória. Muitos usuários ficaram surpresos ao tentarem acessar determinados sites e se depararem com uma página informando que o site estava bloqueado. A Figura 3 ilustra a página que era apresentada.

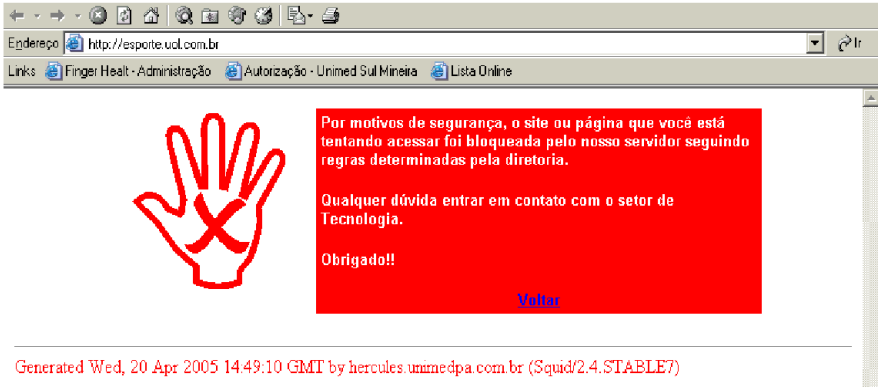


Figura 3: Página exibida quando um site é bloqueado

No mesmo dia que iniciou o bloqueio dos sites, foi realizada uma reunião com os responsáveis pelos setores explicando que o acesso a internet estava sendo monitorado e controlado, que todos deveriam relacionar os *sites* que costumam trabalhar e que a partir daquele momento novos *sites* que deveriam ser liberado a todos colaboradores deveriam passar pelo setor de tecnologia da informação.

Os resultados alcançados não podiam ser melhores. Os relatórios que no início apresentavam inúmeros acessos bloqueados foram diminuindo com o tempo. Os relatórios continuaram a ser analisados, não mais diariamene, mas semanalmente e por amostragem, mantendo a lista negra de *sites* que deveriam ser bloqueados sempre atualizada. Problemas com vírus que vinham nas mensagens recebidas através de sistemas de webmail, acessados para a consulta de *e-mails* particulares, se extinguiram com o bloqueio dos respectivos *sites*, minimizando os problemas com suporte.

Um dos reflexos diretos que foi observado pode ser exposto através do gráfico apresentando na Figura 4, fornecido pela operadora de telecomunicações[4]. Até o início do mês de janeiro o tráfego no *link* era pequeno e se mantinha dentro de um mesmo patamar, o que é justificado pela pouca popularidade da internet entre os colaboradores. Mas com a familiaridade com os novos recursos e a nova tecnologia, o acesso foi aumentando e mantendo um alto patamar de utilização do *link*. Após a implantação do controle de acesso a

[3] *Blacklist* de *sites*: `http://www.squidguard.org/blacklist/`

[4] O tráfego de saída apresentado no gráfico refere-se à porta de saída do roteador da operadora de telecomunicações para o roteador da empresa, portanto indica o tráfego de entrada no roteador da empresa.

utilização do *link* voltou a ficar em patamares bem menores, como pode ser obeservado no gráfico a partir do final do mês de março.

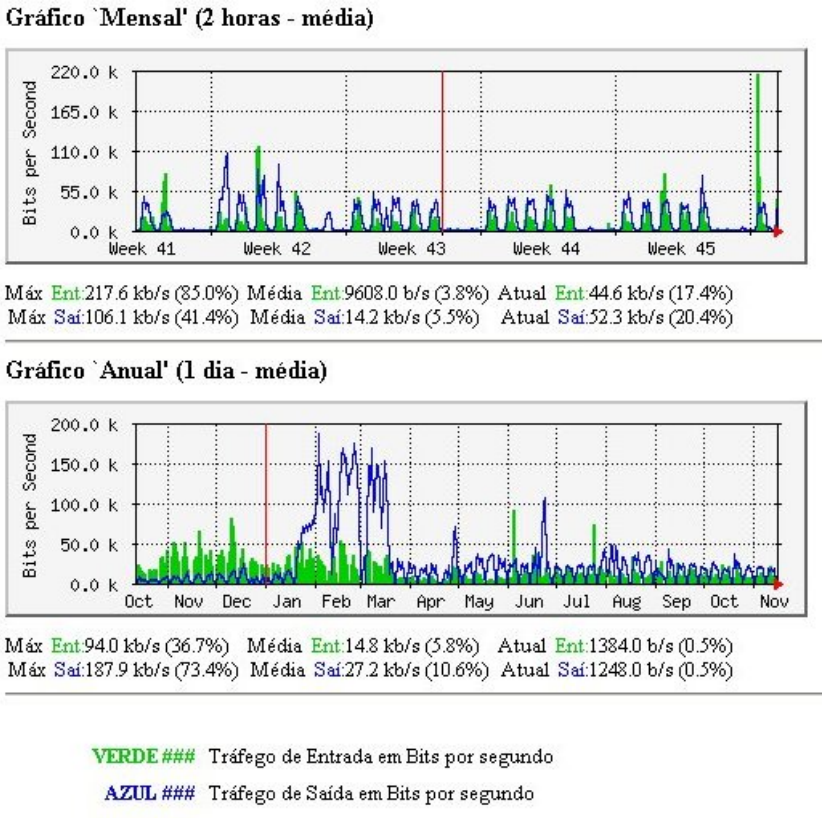


Figura 4: Gráfico de utilização do link com a internet

Estes fatos comprovam que antes do controle de acesso, a utilização dos recursos era feita indiscriminadamente e não apenas para as finalidades a que se destinavam, o que acabava gerando transtornos e custos indiretos, como lentidão no *link*, chamados de suporte entre outros. Além disso, os resultados obtidos comprovaram a eficácia do Squid para controle de acesso à internet.

Referências

ANKLESARIA, F.; MCCAHILL, M.; LINDNER, P.; JOHNSON, D.; TORREY, D.; ALBERTI, B. *The Internet Gopher Protocol (a distributed document search and retrieval protocol)*. Internet Engineering Task Force (IETF), March 1993. (Request for Comments: 959). Disponível em: <http://www.ietf.org/>.

CASE, J.; FEDOR, M.; SCHOFFSTALL, M.; DAVIN, J. *A Simple Network Management*

Protocol (SNMP). Internet Engineering Task Force (IETF), May 1990. (Request for Comments: 959). Disponível em: <http://www.ietf.org/>.

FIELDING, R.; GETTYS, J.; MOGUL, J. C.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. *Hypertext Transfer Protocol – HTTP/1.1*. Internet Engineering Task Force (IETF), June 1999. (Request for Comments: 2616). Disponível em: <http://www.ietf.org/>.

FONSECA, E. L. S. Análise de desempenho de servidores proxy cache www. In: *SPG'98 - II Semana de Pós-Graduação em Ciência da Computação*. Belo Horizonte: DCC/UFMG, 1998. Disponível em: <http://www.dcc.ufmg.br/pos/html/spg98% -/anais/erik/erik.html>.

FREE SOFTWARE FOUNDATION. *GNU General Public Licence Version 2*. June 1991. Disponível em: <http://www.gnu.org/licenses/gpl.html>.

JARGAS, A. M. *Expressões Regulares – Guia de Consulta Rápida*. São Paulo: Novatec, 2001.

JUCÁ, H. L. *Implementação de Firewall em Linux*. São Paulo: Brasport, 2003.

NIC BR SECURITY OFFICE. *Práticas de Seguranca para Administradores de Redes Internet, Versão 1.2*. [S.l.], 16 de Maio 2003. Disponível em: <http://www.nbso.nic.br-/docs/seg-adm-redes/seg-adm-redes.pdf>.

POSTEL, J.; REYNOLDS, J. *File Transfer Protocol*. Internet Engineering Task Force (IETF), October 1985. (Request for Comments: 959). Disponível em: <http://www.ietf.org/>.

RUFINO, N. M. de O. *Técnicas e Ferramentas de Ataque e Defesa de Redes de Computadores*. São Paulo: Novatec, 2002.

SRISURESH, P.; EGEVANG, K. *Traditional IP Network Address Translator (Traditional NAT)*. Internet Engineering Task Force (IETF), January 2001. (Request for Comments: 3022). Disponível em: <http://www.ietf.org/>.

UCHÔA, J. Q. *Segurança em Redes e Criptografia*. Lavras: UFLA/FAEPE, 2003. (Curso de Pós Graduação "Lato Sensu" (Especialização) a Distância em Administração em Redes Linux).

UCHÔA, J. Q.; SIMEONE, L. E.; SICA, F. C. *Administração de Redes Linux*. Lavras: UFLA/FAEPE, 2003. (Curso de Pós Graduação "Lato Sensu" (Especialização) a Distância em Administração em Redes Linux).

VISOLVE.COM. *Squid Configuration Manual*. [S.l.], May 2002. Disponível em: <http://squid.visolve.com/squid/squid24s1/squid24s1.pdf>.

WESSELS, D.; CLAFFY, K. *Internet Cache Protocol (ICP), Version 2*. Internet Engineering Task Force (IETF), September 1997. (Request for Comments: 959). Disponível em: <http://www.ietf.org/>.