

Proxy

O Termo “servidor proxy”, vem de uma palavra em inglês que significa procuração. Este aplicativo também pode ser conhecido como gateway de aplicação, entretanto ficando limitado apenas a alguns protocolos como http, https e ftp. Hoje o proxy tem a grande vantagem de atuar como um cache de páginas web. Se você abre, por exemplo o site www.rodolfoavelino.com.br, na segunda vez que acessá-lo, tudo que estiver armazenado no espaço em disco do proxy será enviado ao seu browser sem que o proxy tenha feito um novo download na internet, economizando tempo e banda de internet.

As requisições de sites são feitas das estações através do proxy, ou seja, o proxy, é o responsável em realizar a busca e entrega do site para o usuário.

Squid

O Squid busca por comunicação TCP e ICP (Internet Cache Protocol), em portas específicas. O TCP é usado para a comunicação entre Webservers e clientes, e o ICP para conversar entre servidores de cache.

O Squid suporta os protocolos HTTP, HTTPS, FTP e outros. Foi escrito originalmente para rodar em sistema operacional tipo Unix, mas ele também funciona em sistemas Windows desde sua versão 2.6.STABLE4.

O Squid é um dos softwares mais utilizados como servidor proxy de web para o sistema operacional Gnu/Linux. O software está presente na maior parte das distribuições Linux, mas também é possível realizar o download e procurar por atualizações no site oficial www.squid-cache.org. As características listadas a seguir são informações que estão disponíveis no mesmo site oficial do Squid, com contribuições da comunidade do Squid aqui no Brasil.

Características do Squid

Além da capacidade de intermediar o acesso à internet, como já mencionado, o Squid tem outros recursos que o torna uma excelente alternativa para aproveitamento mais racional da comunicação. Dentre esses recursos podemos destacar:

Cache - através desse recurso o Squid armazena em cache o conteúdo acessado, de forma que se algum host fizer novamente uma requisição ao mesmo conteúdo, que já se encontra armazenado, ele recebe diretamente do cache, sem a necessidade de efetuar uma nova busca dos dados na internet. O uso desse recurso pode trazer uma rapidez maior ao acesso à internet, pois provavelmente o link do host com o proxy é bem mais rápido do que deste com a internet;

Autenticação - podemos restringir o acesso ao servidor proxy com o uso da autenticação de usuários, de forma que seja melhorada a segurança, já que somente usuários autorizados poderão acessar a internet. Este recurso é bastante flexível e pode ser implementado de várias maneiras, como uso do protocolo LDAP, SMB, módulos PAM, etc;

Registro de acessos - os acessos são registrados em arquivos de log, podendo esses serem utilizados para as mais diversas finalidades, que vão desde a análise de performance do servidor, até a geração de relatórios detalhados dos acessos à internet. Existem vários softwares analisadores de logs do Squid capazes de gerar relatórios tão bons, que por si já justificariam o uso do Squid, em razão do controle proporcionado;

Controle centralizado - com o uso do proxy temos a facilidade de um único ponto centralizador do acesso à internet, o que torna a gerência da rede mais fácil e eficiente. Uma única máquina é capaz de prover acesso à várias outras;

Segurança - como apenas o proxy está diretamente ligado à internet, temos apenas uma (ou mesmo poucas, caso tenhamos mais de um servidor proxy) máquina potencialmente vulnerável. Desta forma fica mais fácil concentrar esforços na melhoria da segurança de apenas um ponto na rede.

Arquivo de configuração

O arquivo de configuração do Squid garante um grau elevado de capacidade de customização deste servidor. Ele é composto por mais de 3000 linhas, entretanto não há com o que se preocupar, pois a maior parte delas se constitui de comentários referentes as funções de cada TAG e como deve-se utilizá-la. De tantos comentários ele parece mais um manual do que um arquivo de configuração.

Podemos ter um servidor configurado e funcional com apenas pouco mais de 30 linhas no arquivo, devemos considerar também que boa parte das opções configuradas como padrão no arquivo squid.conf não precisam ser alteradas, pois já representam a configuração mais adequada.

Desta forma vamos nos ater as configurações mais importantes e necessárias para que possamos ter um servidor proxy funcional. Abaixo mostraremos algumas destas opções, não querendo tornar isso um guia definitivo para configuração do Squid, mas sim um ponto de partida para implementação de um servidor proxy.

hierarchy_stoplist cgi-bin ?

- Esta opção vem habilitada por padrão no squid.conf e inclusive é recomendado o seu uso. Ela é responsável por dizer ao Squid que ele deve buscar os dados diretamente na origem, sem passar pelos vizinhos na hierarquia. Esta configuração se refere a conteúdo dinâmico, portanto se a URL contém o padrão

aqui especificado, será tomada a atitude de ir direto a origem buscar o conteúdo;

acl QUERY urlpath_regex cgi-bin ?

no_cache deny QUERY

- Esta ACL diz ao squid para não armazenar em cache o conteúdo dos CGI's, pois obviamente não é interessante por tratar-se de conteúdo dinâmico;

cache_mem 128 MB

- Nesta opção é definida a quantidade de memória que o Squid irá usar, mas é bom lembrar que o manual do Squid adverte que a memória aqui mencionada é referente a objetos em trânsito, objetos “quentes” e objetos com negativa de cache, ou seja, é a memória apenas para ele utilizar na manipulação dos seus objetos e não ao total de memória consumida por ele, que pode ser duas ou três vezes maior;

cache_dir ufs /var/cache/squid 300 64 64

- Esta TAG determina onde e como será feito o cache, neste caso o diretório de cache é /var/cache/squid, com o tamanho de 300 MB, tendo 64 diretórios com 64 subdiretórios dentro de cada um deles;

auth_param basic program /usr/lib/squid/nsc_auth /etc/squid/passwd

auth_param basic children 5

auth_param basic realm Servidor Proxy ARL

auth_param basic credentialsttl 2 hours

- TAG's referentes ao processo de autenticação. Estes parâmetros só serão utilizados caso você opte por ter o acesso a internet com autenticação. Se estes parâmetros estiverem comentados, o acesso não será autenticado. Existem outros métodos de autenticação, podendo ser transparentes ou não ao usuário.

refresh_pattern ^ftp: 1440 20% 10080

refresh_pattern ^gopher: 1440 0% 1440

refresh_pattern . 0 20% 4320

- Estas opções são o padrão do Squid e configuram como serão tratados os tempos de vida dos objetos no cache, isto é, o Squid faz uso destes valores para verificar se os objetos armazenados são os mais recentes ou há a necessidade de atualizá-los. Não há necessidade de mudança nestes valores;

acl all src 0.0.0.0/0.0.0.0

acl manager proto cache_object

acl localhost src 127.0.0.1/255.255.255.255

acl to_localhost dst 127.0.0.0/8

acl SSL_ports port 443 563 acl Safe_ports port 80 # http

acl Safe_ports port 21 # ftp

acl Safe_ports port 443 563 # https, snews

acl Safe_ports port 70 # gopher

acl Safe_ports port 210 # wais

acl Safe_ports port 1025-65535 # unregistered ports

acl Safe_ports port 280 # http-mgmt

acl Safe_ports port 488 # gss-http

acl Safe_ports port 591 # filemaker

acl Safe_ports port 777 # multiling http

acl CONNECT method CONNECT

- Estas ACL's fazem parte da configuração padrão do Squid e é o mínimo recomendável para seu uso, não sendo necessária nenhuma alteração nas mesmas;

acl MINHA_REDE src 192.168.16.0/24

acl USUARIOS proxy_auth REQUIRED # quando os parametros de autenticação estiverem habilitados

acl BLOQUEADOS url_regex -i "/etc/squid/sites_bloqueados"

acl BLOQUEADOS1 url_regex -i "/etc/squid/sites_bloqueados1"

acl NAO_BLOQUEADOS url_regex -i "/etc/squid/sites_liberados"

- Seção do arquivo com ACL's definidas pelo administrador, além das anteriormente comentadas que já vêm configuradas por padrão.

http_access allow manager localhost # dá acesso ao protocolo cache_object apenas para o próprio servidor

http_access deny manager # nega o acesso ao protocolo cache_object para qualquer outra máquina

http_access deny !Safe_ports # nega acesso a qualquer outra porta além das definidas na acl Safe_ports

http_access deny CONNECT !SSL_ports # nega qualquer conexão que não seja referente às portas seguras

- Definição de regras de acesso referentes as ACL's da parte da configuração padrão do Squid, também não é necessária nenhuma alteração, portanto apenas acrescente as suas próprias regras a estas;

http_access allow USUARIOS NAO_BLOQUEADOS MINHA_REDE

http_access allow USUARIOS !BLOQUEADOS !BLOQUEADOS1 MINHA_REDE

- Estas regras de acesso foram feitas com o uso das ACL's criadas anteriormente na área designada ao usuário, como podemos observar elas foram feitas pelo próprio usuário;

http_access deny all

- Esta regra de acesso é recomendada para uso como última regra da lista define o acesso ao proxy. Ela diz ao Squid que se nenhuma das regras anteriores for aplicada o acesso será então negado, portanto seu uso vai impedir acessos indesejados ao proxy. É muito importante que esta seja a última regra da sua

lista, pois o Squid tem uma regra implícita que inverte a última regra da lista caso nenhuma delas seja aplicada. Isto pode trazer consequências indesejáveis principalmente se for usado negação ou liberação explícita como por exemplo `http_access deny BLOQUEADOS` ou `http_access allow NAO_BLOQUEADOS`;

icp_access allow all

- Permite o acesso a porta icp de acordo com a configuração feita na ACL all, que no nosso caso representa qualquer origem. Esta porta é usada para troca de informações entre servidores proxy;

cache_mgr email_administrador

- Opção para configuração do e-mail do administrador do proxy que vai aparecer nas mensagens de erro. Isto é importante para que os usuários tenham informações de como interagir com o responsável pelo servidor em caso de problemas, como por exemplo um acesso bloqueado de forma errada;

visible_hostname proxy.rodolfoavelino.com.br

- Mostra o nome do servidor configurado nas mensagens de erro, caso contrário o Squid vai tentar descobrir o nome. É indicado ser registrado o FQDN do servidor e não apenas o hostname. Esta informação aparecerá nas telas de erro, no browser das estações.

error_directory /usr/share/squid/errors/Portuguese

- Com o uso dessa opção podemos determinar em qual linguagem serão apresentadas as mensagens de erro. Seu uso é recomendado, pois por padrão as mensagens são em inglês. Em `/usr/share/squid/errors/` existem vários subdiretórios com as linguagens suportadas pelo Squid. E estas mensagens, inclusive, podem ser facilmente personalizadas, pois estão em arquivos no

formato html;

coredump_dir **/var**/cache/squid

- Local onde o Squid gravará seus arquivos *core* em caso de falhas. Estes arquivos são comuns em sistemas UNIX e representam o estado do *software* no momento em que o erro que provocou sua finalização ocorreu.

Algumas outras opções também são interessantes, como no caso das apresentadas abaixo para serem usadas quando estamos com o recurso de proxy transparente habilitado. Estas opções vão fazer com que o Squid se comporte como um servidor web, de forma que o cliente não perceba que está ``conversando" com um proxy.

httpd_accel_port 80

httpd_accel_host [virtual](#)

httpd_accel_with_proxy on

httpd_accel_uses_host_header on

Existem várias outras opções interessantes para serem exploradas, entretanto como não é nosso objetivo aqui dissecar cada uma delas, vamos deixar isso como proposta para uma outra oportunidade.

ACL's

As ACL's - (Access Control Lists) ou listas de controle de acesso, constituem-se na grande flexibilidade e eficiência do Squid, é através delas que podemos criar regras para controlar o acesso à internet das mais diferentes formas. Praticamente todo o processo de controle do Squid é feito com o seu uso.

O uso das listas de controle de acesso é a parte mais importante da configuração de um servidor proxy Squid, pois se bem configuradas podem trazer um nível de segurança muito bom para a rede, entretanto se mau configuradas podem ter

resultado oposto, já que além da falsa sensação de segurança não será aproveitada a grande capacidade e funcionalidade do Squid. Uma dica para se fazer boas ACL's é testar, testar e testar, e depois quando achar que está bom, continue testando e monitorando seus logs para identificar algum problema que possa ocorrer.

A primeira coisa que devemos saber é que o Squid interpreta as ACL's de cima para baixo, portanto é muito importante que seja observada esta regra no momento em que estão sendo construídas as regras de acesso. As ACL's são *case-sensitive*, isto é, `site_bloqueado` é diferente de `Site_bloqueado`, portanto cuidado com isso. Caso seja usada a opção `-i` elas deixarão de ser *case-sensitive*.

Uma outra regra muito importante é que caso uma das regras coincida, as demais não serão mais verificadas. Isso independe da quantidade de regras que ainda faltam para atingir o fim da lista.

Tipos de ACL's

As ACL's são definidas da seguinte forma:

```
acl nome tipo string | ``arquivo"
```

Existem vários tipos de ACL que podemos utilizar, abaixo temos os mais comuns:

src - tipo utilizado para indicar endereços IP de origem. Pode-se especificar um endereço de rede, como `192.168.16.0/24`, um endereço de um determinado *host*, como `192.168.16.10/24` ou uma faixa de endereços, como `192.168.16.10-192.168.16.20/24`;

dst - semelhante ao tipo anterior, mas está relacionada ao endereço de destino;

srcdomain - tipo indicado para verificar o domínio da máquina cliente. Os domínios serão obtidos por resolução reversa de IP, o que pode causar atrasos para a resposta da requisição. A definição do domínio deve ser feita da seguinte

forma: ``.meudominio.com.br", não podendo ser esquecido o ``." (ponto) no início;

dstdomain - usado da mesma forma que *srcdomain*, entretanto com relação ao destino;

srcdom_regex - avalia o domínio usando expressões regulares. Seu uso é semelhante as duas anteriores, acrescentando a flexibilidade do uso da expressão regular;

dstdom_regex - usado da mesma forma que *srcdom_regex*, entretanto com relação ao destino;

time - usado para especificar dias da semana e horários. Os dias da semana são definidos através de letras que os representam, e os horários através de intervalos na forma hora:minuto_inicio-hora:minuto_final. Os dias da semana são especificados assim: S - Sunday (Domingo), M - Monday (Segunda-Feira), T - Tuesday (Terça-Feira), W - Wednesday (Quarta-Feira), H - Thursday (Quinta-Feira), F - Friday (Sexta-Feira) e A - Saturday (Sábado);

url_regex - Este tipo percorre a URL a procura da expressão regular especificada. Deve ser observado que a expressão é *case-sensitive*, para que seja *case-insensitive* deve ser usada a opção -i. É o tipo mais comum de ACL dada a flexibilidade proporcionada pelo uso de expressões regulares;

urpath_regex - Tipo semelhante a *url_regex*, mas procura a expressão regular na URL sem levar em conta o nome do servidor e o protocolo, isto quer dizer que a procura vai ser feita apenas na parte da URL após o nome do servidor, como por exemplo, na URL <http://www.servidor.com.br/pasta/sexo.html> a procura será realizada apenas na parte */pasta/sexo.html*. Ela é também *case-sensitive*, para que seja *case-insensitive* deve ser usada a opção -i;

port - Realiza o controle pela porta de destino do servidor, neste tipo deve ser

especificado o número da porta;

proto - Serve para especificar o protocolo, como por exemplo FTP ou HTTP;

method - Especifica o tipo de método usado na requisição, como por exemplo GET, CONNECT ou POST;

browser - Usa uma expressão regular para tentar ``casar" com os dados do cabeçalho HTTP e combinando então com o navegador utilizado pelo cliente;

ident - Realiza o controle de acesso baseado no nome do usuário. Este tipo requer um servidor Ident rodando na máquina do cliente;

ident_regex - Semelhante a ident, mas utilizando expressão regular;

proxy_auth - Tipo usado para implementar autenticação de usuários no proxy. A autenticação é feita com uso de *softwares* externos. Podem ser passados os nomes dos usuários ou usada a opção REQUIRED para que seja autenticado qualquer usuário válido;

snmp_community - Tipo usado para especificar o nome da comunidade SNMP, para que se possa monitorar o Squid através deste protocolo;

maxconn - Especifica um limite de conexões vindas de um determinado cliente, interessante para uso com outras ACL's de forma a limitar quantidades de conexões para determinados endereços específicos;

req_mime_type - Especifica uma expressão regular para ser verificada no cabeçalho da requisição em busca de um tipo MIME que coincida com o especificado;

arp - Tipo usado para construir lista de acesso baseada no *MAC Address* da interface de rede do cliente, ou seja, em vez de endereço IP da placa, usa-se o seu endereço MAC.

Construindo regras de acesso

Muitas são as possibilidades de construção de listas de acesso com o uso destes e outros tipos citados acima, mas o fato é que para cada situação específica vai ser necessário o uso de uma ou mais ACL's, de um mesmo tipo ou de tipos diferentes, de forma que possa se chegar a uma situação ideal de controle. Não existe uma regra rígida para criação de listas de acesso, portanto vai depender muito da necessidade e habilidade de cada um.

Vamos descrever algumas situações de como podemos construir listas de controle de acesso para atender a cada uma das situações propostas, observando que se trata de uma das formas de resolver o problema e não a única. É interessante observar que o controle de acesso deve ser construído em duas etapas, primeiramente são feitas as classes de ACL's e depois são construídas as regras de acesso com o uso destas ACL's, com o uso de operadores.

Existem vários operadores, dos quais o mais usado é o **http_access**. Com ele podemos construir regras de controle de acesso baseadas no protocolo HTTP e FTP. Dado ao fato deste ser o mais importante operador, vamos focar o nosso trabalho no seu uso.

No momento da construção das regras devemos levar em consideração além das observações que já foram feitas o fato de que se nenhuma das regras forem aplicadas, o Squid aplica o oposto da última regra da lista, portanto é muito importante que seja incluída como última linha das regras de acesso `http_access deny all`". Desta forma não vamos deixar que esta inversão de regra possa provocar algum dissabor, pois negaremos qualquer acesso que não tenha uma regra específica.

Apresentamos abaixo alguns exemplos de como construir regras de acesso:

1. Controlando acesso pela origem:

```
acl localnet src 192.168.16.0/24
http_access allow localnet
http_access deny all
```

Com estas regras estamos especificando que requisições vindas da rede 192.168.16.0/24 serão aceitas, nos demais casos o acesso será negado;

2. Controlando o acesso pelo destino:

```
acl outra_rede dst 192.168.17.0/24
http_access allow outra_rede
http_access deny all
```

Com estas regras estamos limitando o acesso apenas à rede 192.168.17.0/24, desta forma qualquer acesso para outro destino será negado;

3. Controlando acesso pela origem e destino:

```
acl localnet src 192.168.16.0/24
acl outra_rede dst 192.168.17.0/24
http_access allow localnet outra_rede
http_access deny all
```

Aqui juntamos as duas anteriores e estamos especificando que requisições vindas da rede 192.168.16.0/24 e que sejam destinadas à rede 192.168.17.0/24 serão aceitas, nos demais casos o acesso será negado;

4. Controlando acesso pela nome de domínio de origem e destino:

```
acl meu_dominio srcdomain .meudominio.com.br
acl mundo_lafora dstdomain .lafora.com.br
http_access allow meu_dominio mundo_lafora
```

```
http_access deny all
```

Estas regras são semelhantes as acima, onde usamos src e dst, entretanto aqui é verificado o domínio, neste caso requisições vindas de .meudominio.com.br e com destino a .lafora.com.br serão liberadas, nos demais casos o acesso será negado. É muito importante não esquecer o ``." (isso mesmo, o ponto) antes do nome do domínio, sob pena da regra não funcionar;

5. **Controlando o acesso usando autenticação de usuário:**

```
acl usuarios proxy_auth REQUIRED  
http_access allow usuarios  
http_access deny all
```

Esta regra faz com que o acesso seja liberado apenas para usuários que sejam válidos, dado ao uso da opção REQUIRED, e que se autenticuem no Squid. Para que a autenticação funcione devem ser configuradas as TAG's que informam ao Squid como esta será feita, já que ela é realizada por *softwares* externos, mas isto já foi visto na seção [5](#);

6. **Controlando acesso pela origem, destino e autenticação de usuário:**

```
acl localnet src 192.168.16.0/24  
acl outra_rede dst 192.168.17.0/24  
acl usuarios proxy_auth REQUIRED  
http_access allow localnet outra_rede usuarios  
http_access deny all
```

Com estas regras estamos especificando que requisições vindas da rede 192.168.16.0/24 e que sejam destinadas à rede 192.168.17.0/24 serão aceitas, mas desde que os usuários sejam autenticados, nos demais casos o acesso será negado;

7. Acrescentando controle de acesso pelo horário:

```
acl localnet src 192.168.16.0/24
acl outra_rede dst 192.168.17.0/24
acl usuarios proxy_auth REQUIRED
acl expediente time MTWHF 8:00-18:00
http_access allow expediente localnet outra_rede usuarios
http_access deny all
```

Apenas acrescentamos o controle baseado no horário. Veja também a ordem em que as ACL's são verificadas, observe que a verificação dos usuários é feita por último, pois não vamos forçar o usuários a se autenticarem para só depois negar o acesso por um outro motivo. Isto não é obrigatório, apenas no nosso caso fica melhor desta forma, pois se, por exemplo, o horário não está dentro do permitido, o acesso será negado imediatamente sem nem mesmo tentar checar as outras ACL's;

8. Controlando o acesso por palavras chaves:

```
acl MINHA_REDE src 192.168.16.0/24
acl USUARIOS proxy_auth REQUIRED
acl BLOQUEADOS url_regex -i "/etc/squid/sites_bloqueados"
acl BLOQUEADOS1 url_regex -i "/etc/squid/sites_bloqueados1"
acl NAO_BLOQUEADOS url_regex -i "/etc/squid/sites_liberados"
http_access allow USUARIOS NAO_BLOQUEADOS MINHA_REDE
http_access allow USUARIOS !BLOQUEADOS !BLOQUEADOS1 MINHA_REDE
http_access deny all
```

Este é sem dúvida o tipo de controle mais usado, pois com ele podemos fazer o bloqueio de conteúdo considerado impróprio de maneira muito fácil. No nosso caso estamos fazendo o uso de arquivos que contém as listas de

palavras ou sites que consideramos inadequados, no caso /etc/squid/sites_bloqueados e /etc/squid/sites_bloqueados1 e outro arquivo com sites e palavras que podem ser barradas por alguma regra dos arquivos anteriores, mas que na verdade não representam conteúdo impróprio, no caso /etc/squid/sites_liberados. Por exemplo, nem sempre a palavra sexo representa necessidade de bloqueio, pois podemos ter sites de conteúdo educativo que contenham essa palavra na sua URL, por exemplo www.ministeriodealgumacoisa.gov.br/sexoseguro.html. As palavras ou sites devem ser colocados no arquivo uma por linha.

Então aqui estamos fazendo o seguinte: primeiramente vamos liberar usuários que desejem acessar sites listados no arquivo /etc/squid/sites_liberados e que pertençam a nossa rede, no caso 192.168.16.0/24 ou usuários que queiram acessar sites com conteúdos diferentes (! significa diferente) do listado nos arquivos /etc/squid/sites_bloqueados e /etc/squid/sites_bloqueados1 e que pertençam a nossa rede, em ambos os casos deverá ser feita a autenticação.

<http://www.squid-cache.org.br/>