Filtering home Internet access with Squid

PARENTAL GUIDANCE

Are your children wearing out their eyeballs on the Internet? Squid will help you impose some time limits and filter out inappropriate content.

BY FLORIAN EFFENBERGER

Corporate networks often use the Squid proxy server to filter Internet traffic. If you are using a Linux system as your home router and firewall, you can also use Squid on a smaller scale to create access rules for your home network. A few simple commands will help you establish a schedule for Internet use and rule out sites with inappropriate content.

Getting Started

These examples assume your Linux home server is acting as a firewall and router. The Linux system will be the only computer on the home network with an Internet connection, and it will act as a proxy server, giving all the other clients access in a way that ensures nobody can work around the filter. For this reason, the router will not forward TCP ports 21 and 80, forcing the clients to access the proxy for the ftp and http protocols.

System Requirements

The information in this article is based on Squid version 2.6 [1]. Debian 4.0 and Ubuntu 8.04 users can simply type *apt-get install squid* to install the proxy.

The network clients can use any operating system that supports TCP/IP. The clients need a browser entry defining the router as the proxy host (Figure 1).

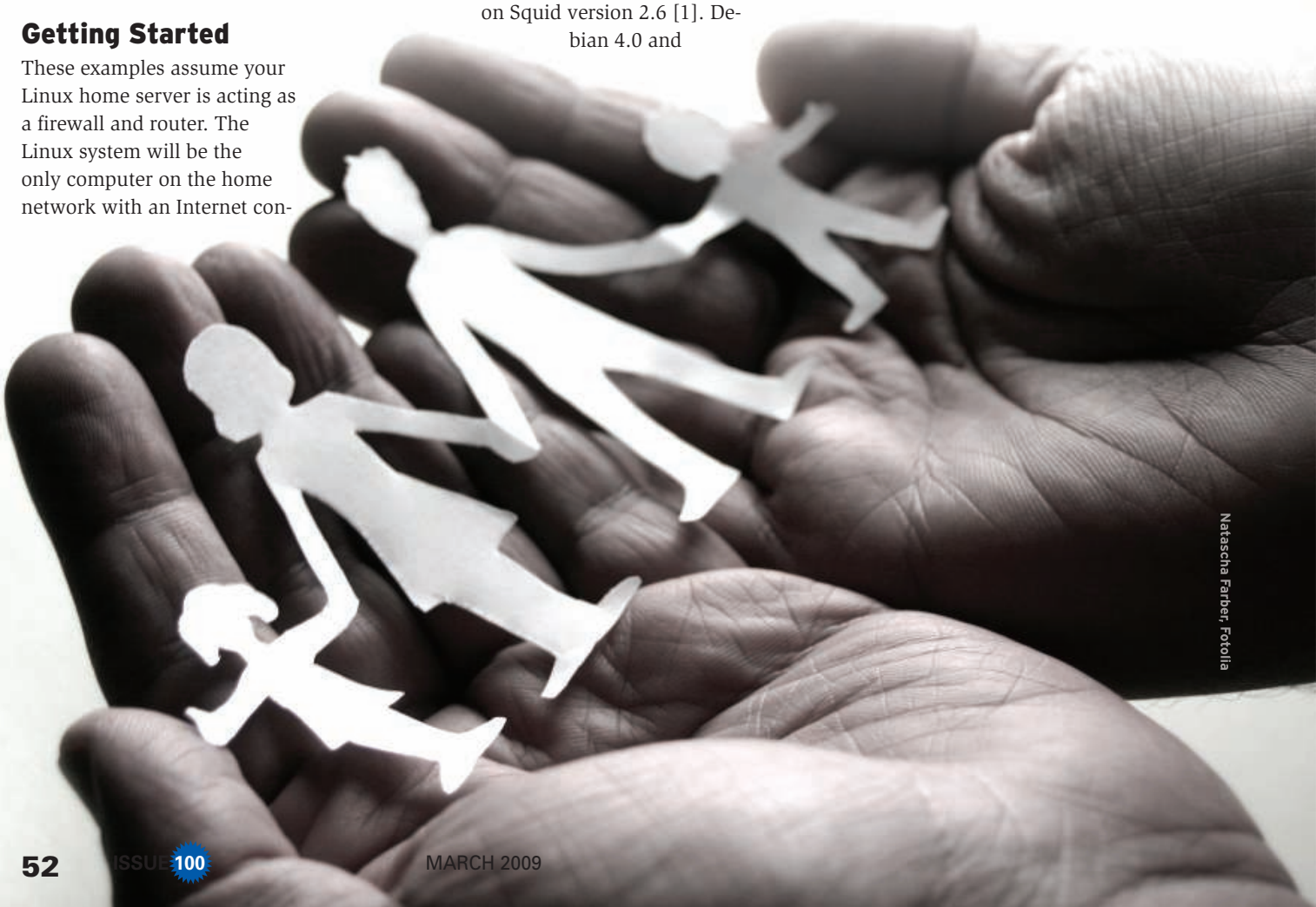
Basic Configuration

The */etc/squid/squid.conf* file is at the center of your Squid configuration. A version of this *squid.conf* file with useful comments is available on the web [2].

The first step is to make sure the firewall blocks incoming requests for the proxy from all external networks. This precaution prevents third parties from using the server for Internet access.

Listing 1 gives two approaches for blocking requests from external networks – make sure you customize these entries to match your own environment. It is a good idea to start by blocking all the ports on the firewall and then explicitly allowing only those ports you really need.

If you apply multiple examples at the same time, the order and the combination of parameters is important. For more details, read the Squid help files. To force the program to parse the updated configuration, just type */etc/init.d/*



Natascha Farber, Fotolia

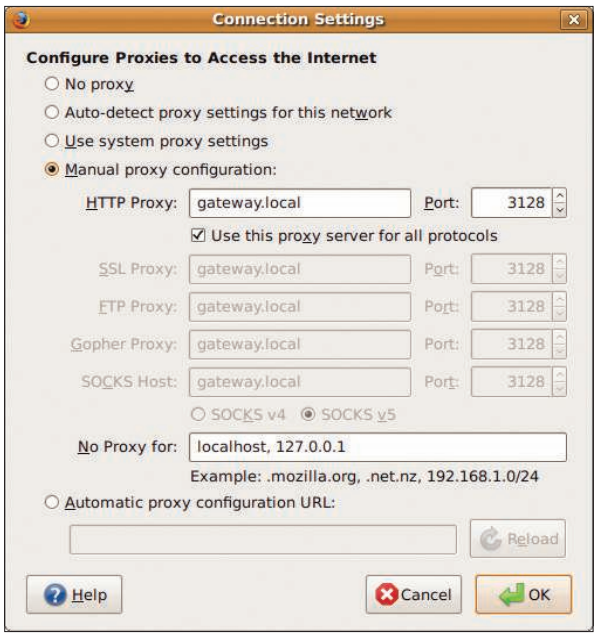


Figure 1: Configuring a proxy host for a web browser.

squid reload at the command line. (See the box titled "Critical Security Information" for some security tips.)

Client Maintenance

The next step is to modify the ACL (Access Control List) settings in *squid.conf*. To assign different filter rules to different users – for example, stricter rules for children – you first need to tell Squid what criteria to apply to incoming requests.

Listing 2 adds client IP addresses for Mom (Marion), Dad (Archie), a son (Simon), and a daughter (Tanja). At the same time, the listing tells Squid to accept requests from the local network.

Time-Based and Manual Blocks

In my experience, it is often hard to keep the kids off the computer, especially if they have Internet access. Squid lets you block online access at certain times of the day. Listing 3 gives an example that defines a time slot for older kids between 1:00pm and 9:00pm Monday through Friday, while restricting access for the younger kids to 1:00pm and 7:00pm.

Because Simon is older, he is allowed to surf later than his little sister, Tanja; this is set in the *http_access* lines, which are read as follows: "The client called Simon is not allowed to surf the Internet, except at the times defined in the ACL *big_kids*." Both kids are allowed unrestricted Internet access on weekends, and no restrictions apply to the parents.

Sometimes it makes sense to completely block Internet access for a client. Just add the contents of Listing 4 to the configuration file. The */usr/share/squid/blocked_clients* file itself only contains the IP addresses and netmasks of the clients you want to block (Listing 5).

A simple shell command is all it takes to add clients to the list. The command

```
echo 192.168.1.3/32 >>  &&
/usr/share/squid/blocked_clients &&
/etc/init.d/squid reload
```

puts Simon on the block list. Typing

```
sed /^192.168.1.3\\/32$/d -i  &&
/usr/share/squid/blocked_clients &&
/etc/init.d/squid reload
```

removes the entry.

Ads and Cookies

In addition to simple website blocking, Squid offers more advanced features: In combination with the free Privoxy [3] tool, it will filter banners and similar elements while you surf the web. To enable Privoxy, just add the lines from Listing 6.

Blacklists

Even if your kids keep to the times that they are allowed to surf the web, you

Listing 1: Blocking External Access

```
01 # Approach 1
02 # drops incoming requests for Squid port 3128,
03 # except for requests from the 192.168.1.* network
04 iptables -I INPUT -p tcp --dport 3128 -s ! 192.168.1.0/24 -j DROP
05
06 # Approach 2
07 # drops incoming requests for Squid port 3128,
08 # except for requests from the eth0 NIC (local network)
09 iptables -I INPUT -p tcp --dport 3128 -i ! eth0 -j DROP
```

Listing 2: ACL Settings

```
# Individual client definitions
acl marion src 192.168.1.1/32
acl archie src 192.168.1.2/32
acl simon src 192.168.1.3/32
acl tanja src 192.168.1.4/32
01 # Allow Squid to accept requests from the local network
02 acl localhost src 192.168.1.0/24
03 acl to_localhost dst 192.168.1.0/24
```

Listing 3: Time Slots

```
acl big_kids time MTWHF 13:00-21:00
acl small_kids time MTWHF 13:00-19:00
http_access deny simon ! big_kids
http_access deny tanja ! small_kids
```

Listing 4: Blocking Access

```
acl blocked_clients src "/usr/share/
squid/blocked_clients"
http_access deny blocked_clients
```

Listing 5: blocked_clients

```
192.168.1.3/32
192.168.1.4/32
```

Listing 6: Enable Privoxy

```
01 # adding Privoxy as a filter
02 cache_peer 127.0.0.1 parent 8118 7
   no-query
03 never_direct allow all
04
05 # Do not route FTP requests via
   Privoxy
06 acl ftp proto FTP
07 always_direct allow ftp
```

Listing 7: Place websites off limits

```
01 # defines a blacklist that applies to all clients except the parent's clients
02 acl blacklist url_regex -i "/usr/share/squid/blacklist"
03 http_access deny blacklist !marion !archie
04
05 # defines a blacklist that additionally applies to Tanja
06 acl blacklist_tanja url_regex -i "/usr/share/squid/blacklist_tanja"
07 http_access deny tanja blacklist_tanja
```

will not want them accessing sites with pornographic or violent content.

To place websites off limits, you just need to add a couple of lines to your Squid configuration file (see Listing 7) then add entries with strings describing the web content you want to block to the */usr/share/squid/blacklist* file (see Listing 8); regular expressions [4] are supported.

Finally, type */etc/init.d/squid reload* to tell the proxy to parse the blacklist.

Custom Blacklists

Of course, Squid will let you assign different blacklists to different users. For example, Simon is allowed to browse online auctions, whereas Tanja is still too young for such things. To set this up, just assign the blacklist in Listing 8 as */usr/share/squid/blacklist_tanja*.

The example blocks pages that contain the prohibited text. To define more precise filters, you can use regular expressions, but don't rely blindly on the list; it makes far more sense to check at regular intervals to see whether it still has the desired effect. And remember that server and file names do change.

Whitelists

Another approach to filtering, and one that is far more strict, is to use whitelists. If you prefer to restrict Tanja's access to just one or a few sites, a whitelist is probably a good idea. Just add the lines in Listing 9 to your Squid configuration and create a whitelist to match. The syntax is similar to that of the blacklist; however, whitelisting can cause problems when a single website references content from many other locations.

To display the complete page, you would need to list these sites explicitly.

Pre-Configured Filters

Many commercial filtering solutions are based on Squid. Although they are not available under free licenses, and even cost money in some cases, the advantage is that they give you daily updated, graduated filter lists that offer excellent protection.

Listing 8: Block web content

```
01 # blocks all pages/domains with the following strings
02 violence.tld
03 actionmovies.domain
04 nude.xyz
05
06 # blocks the address http://(www.)mailorder.co/orders/,
07 # but grants access to the site otherwise
08 mailorder.co/orders/
09
10 # Prevents downloading of files with the suffixes .mp3 or .exe
11 .mp3
12 .exe
```

Listing 9: Adding a Whitelist

```
01 # Tanja is only allowed to access these pages
02 acl whitelist url_regex -i "/usr/share/squid/whitelist"
03 http_access deny tanja ! whitelist
```

Free filter lists such as the SquidGuard [5] lists are also available.

Conclusions

The Squid proxy server system lets you keep control of Internet traffic – even on a small, home-sized network. If necessary, Squid will act as a transparent proxy that you do not need to configure explicitly on the browser side. Squid also offers simple traffic shaping – that is, the ability to assign bandwidth to clients based on a set of rules. ■

Critical Security Information

The proxy configuration will depend on your network structure. This article is simply intended as a guide, and it assumes a working configuration with the Ubuntu 8.04 packages, including Squid version 2.6.STABLE18. Other distributions or versions might require a different approach.

Before you boot the system, it makes sense to check out the Squid manual and ensure that your system really is secure. An open and incorrectly configured proxy is just as bad as an unprotected WLAN: It would theoretically allow any Internet user to surf the web with the proxy owner's identity. The proxy owner would then be liable for anything these unauthorized users did. A working firewall, up-to-date packages, and a secure Squid configuration are thus mandatory.

INFO

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

[2] Squid configuration file: *http://www.squid-cache.org/Versions/v2/2.6/cfgman/*

[3] Privoxy: *http://www.privoxy.org/*

[4] Regular expressions: *http://en.wikipedia.org/wiki/Regular_Expression*

[5] SquidGuard: *http://www.squidguard.org/blacklists.html*

THE AUTHOR

Florian Effenberger is the OpenOffice.org international Marketing Project Co-Lead and marketing contact for German-speaking countries. He is a member of the NGO OpenOffice.org Germany. In addition to his work with OpenOffice, Florian specializes in designing and managing open source--based school networks.