SquidGuard - Protecting your users and your network
Author - Nic Maurel

Have you ever wanted to control what your users visit and what they don't. Well if you're anything like me I get quite nervous if I have to monitor certain squid access logs. Why go through the pain worry and anguish when you can install SquidGuard, a plugin for squid that controls access to specified websites, at certain times and even certain ip ranges. SquidGuard uses Squid's standard redirector interface so no patching of Squid is needed, thus it is extremely fast and easy to use. SquidGuard is freely distributable and can be downloaded at www.SquidGuard.org. In the following paragraphs I will talk you through what you will need and how to implement SquidGuard.

Step 1
Prerequisites are :

- A linux server with squid already installed. I am using squid version 2.4.STABLE7 (default version with Redhat 8)

Step 2 - Install SquidGuard
Unpack your version of SquidGuard into a directory of your preference..

```
# mkdir  /root/squidsrc (or wherever you like)
# cd /root/squidsrc
# gzip -dc /root/squidsrc/SquidGuard-1.2.0.tar.gz | tar xvf -
# cd SquidGuard-1.2.0
#./configure <---- Make sure for this command you have a gnu gcc compiler installed
```

the default configure will install SquidGuard at */usr/local/SquidGuard* you can specify elsewhere (*--prefix=/some/where/else option*).
For this exercise I will use defaults.

```
#make
#make install
```

This can all be done while squid is running, there is no need to stop squid. The SquidGuard binary will be installed at */usr/local/bin* and the configuration file will be installed at */usr/local/bin/SquidGuard*.

Step 3 - Configuring SquidGuard
As mentioned above SquidGuard has a *SquidGuard.conf* file which is situated in */usr/local/SquidGuard/*, this is what you use to tell SquidGuard what your users will and will not access. Here is an example of a simple configuration file:

```
logdir /usr/local/SquidGuard/log
dbhome /usr/local/SquidGuard/db

dest ads {
        domainlist ads/domains
        urllist    ads/urls
    }

dest warez {
        domainlist warez/domains
        urllist    warez/urls
    expressionlist warez/expressions
    }
```

```
acl {
        default {
           pass !ads !warez all
           redirect http://localhost/cgi/blocked?
clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=%s&url=%u
        }
```

Fairly simple and straight forward I will explain what each of the options mean..

```
logdir <-- specifies your logs for SquidGuard (helpful for troubleshooting)
```
dbhome <-- this is the home for your blacklists

Community blacklists can be downloaded directly from SquidGuard.org and are preformatted and ready to use. You can also contribute to this list or you can create your own list. To create your own all you do is create a file and start adding domains or urls line by line depending on file you are creating here is an example for a domain file

```
# vi domain
195.82.124.99
213.161.20.99
udv.org.br
glasspipes.cc
```

and save the file to the specified place. Moving on, pointing to the db files...

dest <-- is a group that you can specify for targets
{} <-- Start and end your group
domainlist <-- stop certain domains eg. warez.com
urllist <-- stop certain paths defined after the domain eg. t69.com/warez
expressionlist <-- stop certain expressions eg. mp3,avi, wav ect.

And finally your acl group will define how you will use your rules and what actions you will take. You can have multiple acls defined but I just specify one for ease of use...

pass < -- declares groups (eg. dest group) that should pass (! is a NOT operator)
redirect < -- defines the alternate rules that have not passed as well as page to redirect to

Once you have finished with your *SquidGuard.conf* file make sure it is saved before you exit.

Step 4 - Configuring squid with a redirector
In order to plug SquidGuard into squid you must edit the */etc/squid/squid.conf* (default directory) in the following way..

```
# vi /etc/squid/squid.conf
redirect_program /usr/local/bin/SquidGuard -c /usr/local/SquidGuard/SquidGuard.conf
redirect_children 4 <-- is a good number to have but you should never need more than 1 child processes
```

Once that's done save your *squid.conf* and do the following

```
# /etc/init.d/squid restart
        - or -
# `which squid` -k reconfigure
```

and Voila! SquidGuard should start with squid every time.

```
# ps -ef
root    29423    1  0 Sep01 ?       00:00:00 squid -D
squid   29426 29423  0 Sep01 ?       00:02:08 (squid) -D
squid   29427 29426  0 Sep01 ?       00:00:40 (SquidGuard) -c /usr/local/SquidGuard/SquidGuard.conf
squid   29428 29426  0 Sep01 ?       00:00:03 (SquidGuard) -c /usr/local/SquidGuard/SquidGuard.conf
squid   29429 29426  0 Sep01 ?       00:00:03 (SquidGuard) -c /usr/local/SquidGuard/SquidGuard.conf
squid   29430 29426  0 Sep01 ?       00:00:03 (SquidGuard) -c /usr/local/SquidGuard/SquidGuard.conf
squid   29431 29426  0 Sep01 ?       00:00:00 (unlinkd)
```

This is a healthy display

Step 5 --- Troubleshooting
SquidGuard always goes into emergency mode and appears to be started even if you have incorrect syntax in the SquidGuard.conf
Open up two sessions to troubleshoot, one to listen on the log file and one to start and stop the service. In one of your sessions you can do the following

```
# tail -f /usr/local/SquidGuard/log/SquidGuard.log
2005-09-01 15:35:21 [29428] init urllist /usr/local/SquidGuard/db/ads/urls
2005-09-01 15:35:21 [29428] init domainlist /usr/local/SquidGuard/db/warez/domains
2005-09-01 15:35:21 [29428] init urllist /usr/local/SquidGuard/db/warez/urls
2005-09-01 15:35:21 [29428] SquidGuard 1.2.0 started (1126186505.225)
2005-09-01 15:35:21 [29428] SquidGuard ready for requests (1126186521.374)
```

If you see SquidGuard ready for requests then your config is just fine. If you see this:

```
2005-09-01 15:12:34 [29170] parse error in configfile /usr/local/SquidGuard/SquidGuard.conf line 60
2005-09-01 15:12:34 [29170] going into emergency mode
```

then you most likely have something wrong with your syntax on line 60. Log files are very useful for troubleshooting and most of the time leads to your solution. So don't get frustrated just refer to logs, or there is loads of documentation on the SquidGuard website.

Conclusion
As server administrators it is our duty to protect users from temptation, or things they could get involved in on questionable websites, balanced against this is the need to ensure that the business resources are used properly without the risk of damaging the businesses name. SquidGuard is one of the ways we can do this, but on the other side of the coin if the blacklists are too stringent we could end up blocking sites that are useful and have some very annoyed users, managers and directors. So be very careful when implementing your blacklists. Enjoy!