Web site blocking techniques

How to use Squid and squidGuard to restrict user access to undesirable web sites

Tedi Heriyanto

For a variety of reasons, organizations have very strict policies regarding web site access. These policies usually mean that not all users have permission to access all web sites.

This article will explain two techniques that can be used to block web site access to specified groups of users at specified times, using Squid's built-in mechanism and using squidGuard.

In this article, the configurations shown are taken from real files that are used by my clients. I have attempted to write the configuration and installation procedures so that they will work with any operating system. However, where there are specific procedures, I only explain how to do things using an RPM-based Linux distribution.

In the last section, I list advantages and disadvantages using each technique. I also mention common problems that you may encounter during the installation and configuration phase.

Introduction

During my work as a Linux consultant, I'm often asked to implement a mechanism that will stop internet users from accessing inappropriate sites, such as sites containing porn and other offensive or inappropriate material.

There are various reasons organizations might want such mechanisms implemented. The main reasons are:

- **Limited bandwidth -** Some of my clients have very limited bandwidth; they are usually connected to the internet using a dial up modem or a leased line, which allow very limited bandwidth (between 56Kbs and 128 Kbps). In such situations, management cannot permit employees to download inappropriate material as it uses up precious bandwidth.
- **Organizational policy** – Many organizations have very strict internet policies regarding offensive material. For this and other reasons, they don't want employees gaining access to inappropriate sites.
- **Working hours** – Many organizations don't want employees to access particular sites during certain hours.

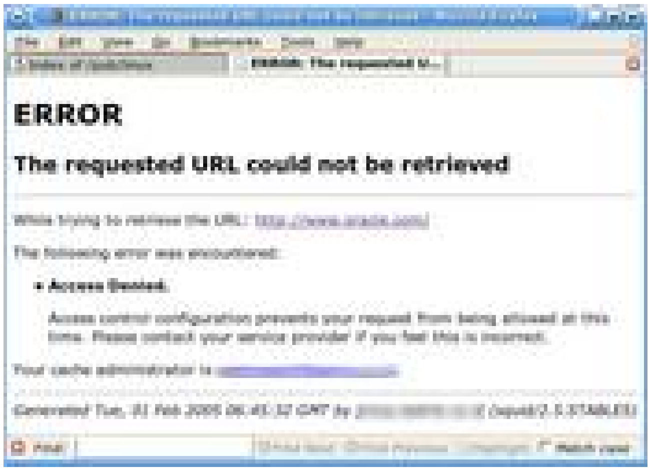
Organizations implement web site blocking policy because of limited bandwidth, organizational policy or working hours

To implement this mechanism, I usually use Squid and squidGuard.

From an ethical point of view, before implementing this kind of policy, the users must be informed about the company's policies - it's even better if they are involved in the policy making process. The organization's internet users, management, and IT (Information Technology) department must define what kind of policy will be implemented.

Before implementing web site blocking policy, you have to ensure it conforms with any legislation.

Testing the Squid blocking mechanism by accessing oracle.com



ERROR

The requested URL could not be retrieved

While trying to retrieve the URL: http://www.oracle.com/

The following error was encountered:

- Access Denied.

  Access control configuration prevents your request from being allowed at this time. Please contact your service provider if you feel this is incorrect.

Your cache administrator is [redacted].

Generated Tue, 01 Feb 2005 06:45:32 GMT by [redacted] (squid/2.5.STABLE5)

Squid is a high-performance, proxy-caching server for web clients, supporting FTP, gopher, and HTTP data objects. Squid can also be used to implement access control.

SquidGuard (http://www.squidguard.org) is a fast and free filter, redirector and access controller for Squid. It was written by Pål Baltzersen and Lars Erik Håland.

In order to use Squid's built-in blocking mechanism, you don't need squidGuard, but you do need Squid to use squidGuard.

Squid's built-in blocking mechanism

In my experience, Squid's built-in blocking mechanism or access control is the easiest method to use for implementing web site blocking policy. All you need to do is modify the Squid configuration file.

Before you can implement web site blocking policy, you have to make sure that you have already installed Squid and that it works. You can consult the Squid web site (http://www.squid-cache.org) to get the latest version of Squid and a guide for installng it.

To deploy the web-site blocking mechanism in Squid, add the following entries to your Squid configuration file (in my system, it's called `squid.conf` and it's located in the `/etc/squid` directory):

```
acl bad url_regex "/etc/squid/squid-block.acl"
http_access deny bad
```

The file `/etc/squid/squid-block.acl` contains web sites or words you want to block. You can name the file whatever you like. If a site has the URL or word listed in `squid-block.acl` file, it won't be accesible to your users. The entries below are found in `squid-block.acl` file used by my clients:

```
.oracle.com
.playboy.com.br
sex
...
```

With the `squid-block.acl` file in action, internet users cannot access the following sites:

- Sites that have addresses ending with .oracle.com
- Sites that have addresses ending with .playboy.com.br
- Sites containing the word "sex" in its pages

In order to use Squid's built-in blocking mechanism, you don't need squidGuard, but you do need Squid to use squidGuard

You should beware that by blocking sites containing the word "sex", you will also block sites such as Middlesex University, Sussex University, etc. To resolve this problem, you can put those sites in a special file called `squid-noblock.acl`:

```
^http://www.middlesex.ac.uk
^http://www.sussex.ac.uk
```

You must also put the "no-block" rule before the "block" rule in the Squid configuration file:

```
...
acl special_urls url_regex "/etc/squid/squid-noblock.acl"
http_access allow admin_ips special_urls

acl bad url_regex "/etc/squid/squid-block.acl"
http_access deny bad
...
```

After editing the ACL files (`squid-block.acl` and `squid-noblock.acl`), you need to restart Squid. If you install the RPM version, usually there is a script in the `/etc/rc.d/init.d` directory to help you manage Squid:

```
# /etc/rc.d/init.d/squid reload
```

To test to see if your Squid blocking mechanism has worked, you can use your browser. Just enter a site whose address is listed on the `squid-block.acl` file in the URL address. In the example above, I block `.oracle.com`, and when I try to access oracle.com, the browser returns an error page.

Using squidGuard blocking mechanism

SquidGuard is:

- Free
- Very flexible
- Very fast
- Easy to install; and
- Portable

You can use squidGuard to:

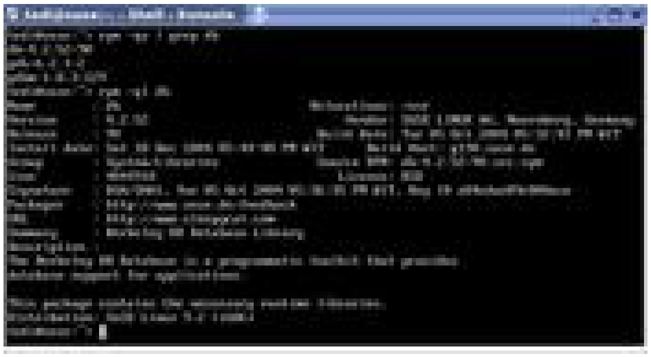
- Limit user web access to a particular web server or URL
- Stop some users from accessing forbidden web servers or URLs
- Block access to URLs matching a regular expression (list or word)
- Forbid the use of IP addresses in URLs
- Redirect blocked URLs to a smart CGI page
- Redirect popular sites to a local mirror
- Have different rules for different times; and
- Have different rules for different user groups

In this article, I will only use squidGuard to block users accessing some sites. Here is how you install squidGuard:

- Install BerkeleyDB version 3.2.9
- Install squidGuard
- Create a squidGuard configuration file that suits your needs
- Create the domain, URL and expression lists you want
- Test squidGuard
- Configure Squid to use squidGuard as the redirector
- Reload Squid

I will describe this procedure in the following sections.

The output of the command “rpm -qa | grep db”



BerkeleyDB and squidGuard installation

Before you can install squidGuard, make sure your system already has BerkeleyDB. You have to match the BerkeleyDB version with the squidGuard version for your system; for example squidGuard version 1.2.0-2 for Red Hat Enterprise Linux 3 needs BerkeleyDB version 4.1.

For RPM-based systems, you can do the following to check if your system has BerkeleyDB or not:

```
$ rpm -qa | grep db
```

If your system doesn’t have BerkeleyDB, you must install it first. Here is the command to install the RPM version:

```
# rpm -ivh db-x.y.z.*.rpm
```

In my experience, the easiest method for installing squidGuard is by using the binary RPM version. A binary, RPM version of squidGuard for a Red Hat based system can be found on Dag Wieers web site (`http://dag.wieers.com/packages/squidguard/`)

SquidGuard configuration and database creation

Before you can use squidGuard, you have to create the squidGuard configuration file and database. You should start the configuration file with only the configuration you need and then extend it later as required.

In my SuSE system, the default path for squidGuard Dag Wieers’ RPM version is `/etc/squid/`

`squidguard.conf`. In other distributions, the configuration file could be `/etc/squid/squidGuard.conf` or `/etc/squidGuard.conf`. The default `squidguard.conf` included with the RPM version can't be used without modification.

Usually my squidGuard configuration file has a structure like this:

- Path declaration
- Source group
- Destination group
- Access control rules

I will only describe the structure used by my configuration. If you want to know more about the squidGuard configuration, please see the file included in the squidGuard package (*configuration.html* or *configuration.txt*, in my system it is in `/usr/share/doc/packages/squidGuard-1.2.0` directory).

> The organization's internet users, management, and IT (Information Technology) department must define what policy the mechanism will implemement

Path Declaration

The path declaration defines the directory for the log files and the database list. For example:

```
dbhome /var/lib/squidGuard/blacklists
logdir /var/log/squidGuard
```

This declaration will set:

- The directory for the squidGuard database list to `/var/lib/squidGuard/blacklists`
- The directory in which to store the log files to `/var/log/squidGuard`

Source Group

Source group has a general syntax like this:

```
src | source name {
    specification
    specification
    ...
}
```

Specification can be any reasonable combination of:

IP addresses and/or ranges (multiple)

- ip xxx.xxx.xxx.xxx […]
- ip xxx.xxx.xxx.xxx/nn […]
- ip xxx.xxx.xxx.xxx/mmm.mmm.mmm.mmm […]
- ip xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy […]

Where:

- xxx.xxx.xxx.xxx is an IP address (host or net, i.e. 10.11.12.13 or 10.11.12.0)
- nn is a net prefix (i.e. /23)
- mmm.mmm.mmm.mmm is a netmask (i.e. 255.255.254.0) and
- yyy.yyy.yyy.yyy is a host address (must be >= xxx.xxx.xxx.xxx)

IP address/range list (single): iplist filename

where:

- filename is either a path relative to dbhome or an absolute path to a database file
- the iplist file format is simply addresses and/or networks separated by a newline as above (without the IP keyword).

For example, the following snippet is in one of my client's squidGuard configuration file:

```
#
# SOURCE ADDRESSES:
#
src admin {
ip       10.10.0.10
}
src users {
    ip       10.10.0.0/255.255.0.0
    ip       10.11.0.0/255.255.0.0
    ip       10.12.0.0/255.255.0.0
}
```

I define two source groups, the *admin group*, which has the IP address 10.10.0.10; and the *users groups*, which have IP addresses in subnetwork of 10.10.0.0, 10.11.0.0 and 10.12.0.0.

Destination Group

Destination group has the following syntax:

```
dest | destination name {
specification
specification
...
}
```

Specification can be any combination of zero or one of each of:

- Domainlist (single):

```
domainlist filename
```

- URL list (single):

```
urllist filename
```

Filename is a text-based database file. You can define the file relative to the database directory path, or you can also define the file absolutely.

Below are the destination groups definitions from my client's configuration file:

```
#
# DESTINATION CLASSES:
#
dest gambling{
log gambling
domainlist gambling/domains
urllist gambling/urls
redirect http://localhost
}
dest warez{
log       warez
domainlist  warez/domains
urllist     warez/urls
redirect http://localhost
}
dest porn{
log       porn
domainlist porn/domains
urllist     porn/urls
redirect http://localhost
}
```

In the configuration above, I define three destination groups: *gambling*, *warez* and *porn*. Their domains and URLs are listed in files called `domains` and `urls` in the directories `gambling`, `warez` and `porn` located in `/var/lib/squidGuard/blacklists/`. When a user tries to access domains and URLs listed in the database, they are redirected to http://localhost.

Access Control List Rules

The Access Control List (ACL) combines the previous definitions into distinct rulesets for each destination and source group:

```
acl {
sourcegroupname {
pass [!]destgroupname [...]
[redirect [301:|302:]new_url]
}
  ...
  default {
pass [!]destgroupname [...]
redirect [301:|302:]new_url
}
}
```

Below is an ACL example:

```
#
# Access Control Lists
#
acl {
admin {
        pass  all
}
users {
pass !gambling !warez !porn all
     redirect 302:http://localhost
}
default {
pass !porn all
     redirect 302:http://localhost
}
}
```

In this configuration:

- The administrators computer has access to every site.
- The user's computers are blocked from gambling, warez and porn domains and URLs listed in the database. When a user tries to access forbidden sites, he or she is redirected to http://localhost
- Computers not listed in the source group are not allowed to access porn domains and URLs. Again, they will be redirected to http://localhost when access porn domains and URLs

SquidGuard database

SquidGuard uses a database that can be divided into an unlimited number of distinct categories like "gambling", "warez", "porn" etc. Each category may consist of separate unlimited lists of domains, and URLs. You can download the blacklists database from here (`http://ftp.teledanmark.no/pub/www/proxy/squidguard/contrib/blacklists.tar.gz`).

Domainlists

The domainlist file has a simple format:

```
domain
domain
...
```

As an example, for a porn category:

```
playboy.com
```

SquidGuard will match any URL with the domain name itself and any sub-domains and hosts (i.e. playboy.com, www.playboy.com, whatever.playboy.com and www.what.ever.playboy.com but not `.*[^.]playboy.com` (i.e. pplayboy.com etc.)).

URLlists

The urllist file has this format:

```
URL
URL
...
```

with the `proto://((www|web|ftp)[0-9]*)?` and `(:port)?` parts and normally also the ending `(/|/[^/]+\.[^/]+)$` part (i.e. ending `/` or `/filename`) taken out (i.e. `http://www3.foo.bar.com:8080/what/ever/index.html` equals `foo.bar.com/what/ever`)

For instance a category for banned sites could be:

```
foo.com/~badguy
bar.com/whatever/suspect
```

All these URLs will match the above urllist:

```
http://foo.com/~badguy
http://foo.com/~badguy/whatever
ftp://foo.com/~badguy/whatever
http://www2.foo.com/~badguy/whatever
http://web56.foo.com/~badguy/whatever
```

but not:

```
http://barfoo.com/~badguy
http://bar.foo.com/~badguy
http://foo.com/~goodguy
```

Configuring Squid to use squidGuard

Because squidGuard is a redirector for Squid, it needs to be called from Squid. Add the following line to the Squid configuration file (`squid.conf`) to instruct Squid to use squidGuard as a redirector:

```
redirect_program /usr/bin/squidGuard -c
  /etc/squidguard.conf
```

Please change `/usr/bin/squidGuard` and `/etc/squidguard.conf` to suit your situation.

Then tell squidGuard to compile its database:

```
# /usr/bin/squidGuard -C all
```

> Sometimes you also need to add a no-block file to allow access to useful sites

Reload Squid

You need to tell Squid to reload its configuration before you can use squidGuard:

- If you install Squid from RPM (SuSE):

```
# rcsquid reload
```

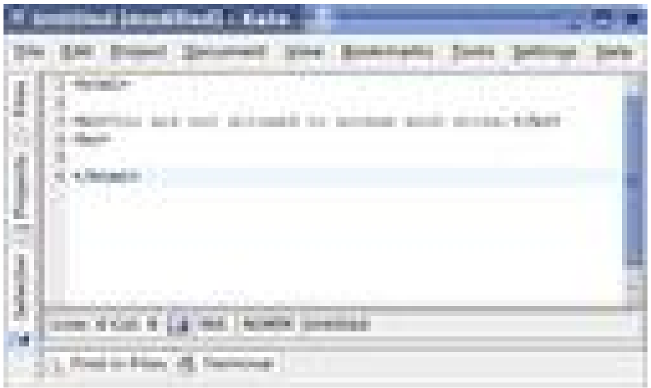
- If you install Squid from RPM (RedHat):

```
# /etc/rc.d/init.d/squid reload
```

- If you install Squid from tarball you can use the following command:

```
# /usr/sbin/squid -k reconfigure
```

Forbidden access page



SquidGuard in action

In this section, I will show a simple and useful squidGuard configuration to block porn sites.

squidguard.conf:

```
logdir /var/log/squidGuard
dbhome /var/lib/squidGuard/db
dest blacklist {
  domainlist blacklist/domains
  urllist  blacklist/urls
  redirect http://localhost
}
acl {
  default {
pass !blacklist all
redirect 302:http://localhost
  }
}
```

blacklist/domains:

```
069palace.com
0ver18.com
1-800-4hotsex.com
1-800-4sex.com
1-8004hotsex.com
1-xxx-live-sex-teen-pussy.nu
10000celebs.com
1000pictures.sinfulxxx.com
1000puresex.com
1001freepics.com
...
playboy.com
```

blacklist/urls:

```
00blow.com/analzone
00blow.com/menonly
0815.org/user/lolitainc
1-800-pussy.com/suzi
1-8004lovers.com/lessie2
1-and-only.com/members-only
100amateurs.com/bondage/images/domination
100amateurs.com/images/toys
100sluts.com/girlie
100teensluts.com/images/xpics
...
```

I redirect the blacklist to an HTML file.

squid.conf:

```
...
redirect_program /usr/sbin/squidGuard -c
  /etc/squidguard.conf
...
acl our_networks src 192.168.0.0/24 127.0.0.1
http_access allow our_networks
```

When I try to access the Playboy web site at www.playboy.com, for example, I get an error message.

Try to access Playboy web site

Transferring data from www.playboy.com...

Closing notes

To implement web site blocking you can use Squid on its own or with squidGuard. Each has advantages and disadvantages.

Squid's built-in mechanism's advantages:

- It's very easy to install and configure
- It's very fast, because it doesn't need external programs
- It's easy to debug. When the blocking mechanism isn't working correctly, you know which program is causing the problem

Unfortunately, it also comes with disadvantages:

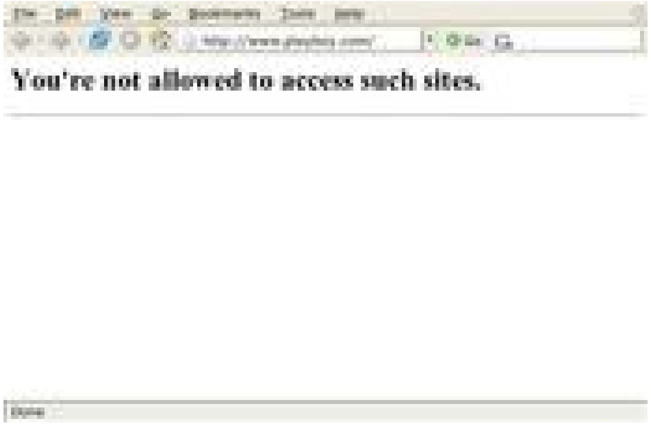
- It's not flexible. It's very hard to configure forbidden domains and URLs categorization.
- Maintainability. When you have several ACLs for blocking sites, it can be a nightmare to edit or modify them to suit your changing needs.

SquidGuard has the following advantages:

- It's very flexible. You can categorize domains and URLs - as many as you like.
- It's maintainable. You can define many ACLs to suit your needs without too much trouble.

But it also comes with some disadvantages:

Error message when accessing Playboy web site



You're not allowed to access such sites.

- It's not easy to install and configure
- It's not very fast because it's called from Squid
- It's harder to debug when there is a problem. In such a case, you have to check the Squid and squidGuard configuration.

You may encounter the following problems during your installation and configuration phase.

Db package not installed

- Install the appropriate version required by squidGuard

Squid is not blocking

- Check the logs (`access.log` and `error.log`)
- Check your `squid.conf` especially the acl rules
- Reload squid

SquidGuard is not blocking

- Check the logs (`squidGuard.log`)
- Check your `squidguard.conf`. If there is an error in this configuration file, squidGuard will not block anything.
- Check the permission for blacklists database. Squid must be able to read the files in the blacklists directory.
- Recompile squidGuard database with `squidGuard -C all`

Bibliography

Baltzersen, Pål and Lars Erik Håland, Configuring squidGuard, /usr/share/doc/packages/squidGuard/doc/configuration.html, 2002

SquidGuard Blacklist (http://www.squidguard.org/blacklist/)

Squid Configuration Guide (http://squid.visolve.com/squid/sqguide.htm)

Squid Documentation Project (http://squid-docs.sourceforge.net/)

Step By Step Install Guide for squidGuard 1.2.0 and BerkeleyDB 3.29 (http://www.maynidea.com/squidguard/step-by-step.html)

SquidGuard is free, very flexible, very fast, easy to install and portable

Copyright information

About the author

During the day, Tedi works as a system engineer and system analyst. He is also a contributing editor for several computer magazines in Indonesia. At night, he works as a computer programmer and security enthusiast. In his previous life, Tedi worked as a software development engineer and as a Linux training instructor.