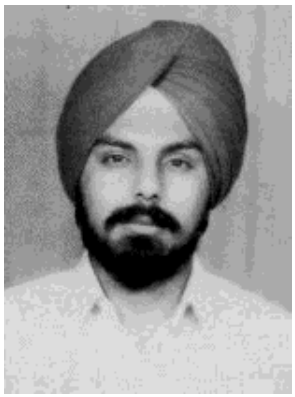
LinuxFocus article number 235
http://linuxfocus.org

Setting up a Squid-Proxy Server



by D.S. Oberoi
<ds_oberoi/at/yahoo.com>

About the author:
D.S. Oberoi lives in Jammu, India and has currently problems to connect to the internet due to the ongoing political tentions.

Abstract:

Linux has become a synonym for Networking. It is being used both in office and home environments as file, print, e-mail, application server and also it is increasingly being used as Proxy server.

A proxy-server provides Internet access to different users at same time i.e. by sharing a single Internet connection. A good proxy server also provides for caching of the requests, which helps to access data from local resources rather fetching the data from web thus reducing access time and bandwidth. Squid is one such software which supports proxy, caching of HTTP, ftp, gopher, etc.. It also supports SSL, access controls, caching of DNS and maintains a full log of all the requests. Squid is as well available for Windows-NT from Logi Sense.

The focus of this article is to give basic guidelines of setting up a proxy server and ways of providing controlled access to users.

Is Squid Installed ?

Squid's rpm comes bundled with the RedHat 7.1 and is installed automatically with the Network OS installation option. One can check whether it is installed or not with the following rpm command:

```
rpm -q squid
```

The latest version of Squid can always be obtained from the Squid Homepage and other mirror sites. Squid can be installed on the desired system by using the following rpm command:

```
rpm -ivh squid-2.3.STABLE4-10.i386.rpm
```

Configuring Squid

The working and behavior of the Squid is controlled by the configuration details given in it's configuration file i.e. squid.conf; this file is usually found in directory the /etc/squid. The configuration file squid.conf is a mile long affair, it just keeps on going for pages after pages, but the good point is that it has all options listed out clearly with explanation.

The first thing that has to be edited is the http_port, which specifies the socket address where the Squid will listen to the client's request; by default this is set to 3128, but can be changed to a user defined value also. Along with the port value, one can also give the IP address of the machine on which Squid is running ; this can be changed to:

http_port 192.168.0.1:8080

With above declaration Squid is bounded to the IP address of 192.168.0.1 and port address of 8080. Any port address can be given; but make sure that no other application is running at set port value. With similar configuration lines other service's request ports can also be set.

Access Control

Through access control features the access to Internet can be controlled in terms of access during particular time interval, caching, access to particular or group of sites, etc.. Squid access control has two different components i.e. ACL elements and access list. An access list infact allows or deny the access to the service.

A few important type of ACL elements are listed below

- src : Source i.e. client's IP addresses
- dst : Destination i.e. server's IP addresses
- srcdomain : Source i.e. client's domain name
- dstdomain : Destination i.e. server's domain name
- time : Time of day and day of week
- url_regex : URL regular expression pattern matching
- urlpath_regex: URL-path regular expression pattern matching, leaves out the protocol and hostname
- proxy_auth : User authentication through external processes
- maxconn : Maximum number of connections limit from a single client IP address

To apply the controls, one has to first define set of ACL and then apply rules on them. The format of an ACL statement is

acl acl_element_name type_of_acl_element values_to_acl

Note :

1. acl_element_name can be any user defined name given to an ACL element.
2. No two ACL elements can have the same name.
3. Each ACL consists of list of values. When checking for a match, the multiple values use OR logic. In other words, an ACL element is matched when any one of its values matches.

4. Not all of the ACL elements can be used with all types of access lists.
5. Different ACL elements are given on different lines and Squid combines them together into one list.

A number of different access lists are available. The ones which we are going to use here are listed below

- **http_access:** Allows HTTP clients to access the HTTP port. This is the primary access control list.
- **no_cache:** Defines the caching of request's responses

An access list rule consists of keywords like allow or deny ; which allows or denies the service to a particular ACL element or to a group of them.
Note:

1. The rules are checked in the order in which they are written and it terminates as soon as rule is matched.
2. An access list can consists of multiple rules.
3. If none of the rules is matched, then the default action is opposite to the last rule in the list; thus it is good to be explicit with the default action.
4. All elements of an access entry are AND'ed together and executed in following manner
   http_access Action statement1 AND statement2 AND statement OR.
   http_access Action statement3
   Multiple http_access statements are OR'ed whereas elements of an access entry are AND'ed together
5. Do remember that rules are always read from top to bottom.

Back to Configuration

By default, Squid will not give any access to clients and access controls have to modified for this purpose. One has to list out one's own rules to allow the access. Scroll down in the squid.conf and enter the following lines just above the http_access deny all line

```
acl mynetwork 192.168.0.1/255.255.255.0
http_access allow mynetwork
```

mynetwork is the acl name and the next line is the rule applicable to a particular acl i.e. mynetwork. 192.168.0.1 refers to the address of the network whose netmask is 255.255.255.0.. mynetwork basically gives a name to group of machines in the network and the following rule allows the access to clients. The above changes along with http_port is good enough to put Squid into gear. After the changes Squid can be started by the following command

```
service squid start
```

Note :
Squid can also be started automatically at boot time by enabling it in ntsysv or setup (System Service Menu). After each and every change in the configuration file, the present Squid process has to be

stopped and for new configuration changes to take effect, Squid has to be started once again. These two steps can be achieved by following commands

1. service squid restart or
2. /etc/rc.d/init.d/squid restart

Client Machine Configuration

Since the client request will be placed at a particular port of the proxy server, client machine's have to be configured for the same purpose. It is taken at this point that these machines are already connected to LAN (with valid IP address) and are able to ping the Linux sever.
For Internet Explorer

1. Go to Tools -> Internet Options
2. Select Connection Tab and click LAN Setting
3. Check Proxy Server box and enter IP address of proxy server and port address where request are being handled (http_port address).

For Netscape Navigator

1. Go to Edit -> Preference -> Advanced -> Proxies.
2. Select Manual Proxy Configuration radio button.
3. Click on View Button &
4. Enter enter IP address of proxy server and port address where request are being handled (http_port address).

Using Access Control

Multiple Access controls and rules offer a very good and flexible way of controlling client's access to Internet. Examples of most commonly used control are given below; this by no means should be taken as the only controls available.

1. Allowing selected machines to have access to the Internet

   ```
   acl allowed_clients src 192.168.0.10 192.168.0.20 192.168.0.30
   http_access allow allowed_clients
   http_access deny !allowed_clients
   ```

   This allows only machine whose IPs are 192.168.0.10, 192.168.0.20 and 192.168.0.30 to have access to Internet and the rest of IP addresses (not listed) are denied the service.
2. Restrict the access during particular duration only

acl allowed_clients src 192.168.0.1/255.255.255.0
acl regular_days time MTWHF 10:00-16:00
http_access allow allowed_clients regular_days
http_access deny allowed_clients

This allows the access to all the clients in network 192.168.0.1 to access the net from Monday to Friday from 10:00am to 4:00 pm.

3. Multipletime access to different clients

acl hosts1 src192.168.0.10
acl hosts2 src 192.168.0.20
acl hosts3 src 192.168.0.30
acl morning time 10:00-13:00
acl lunch time 13:30-14:30
acl evening time 15:00-18:00
http_access allow host1 morning
http_access allow host1 evening
http_access allow host2 lunch
http_access allow host3 evening
http_access deny all

The above rule will allow host1 access during both morning as well as evening hours; where as host2 and host3 will be allowed access only during lunch and evening hours respectively.

Note:
All elements of an access entry are AND'ed together and executed in following manner

http_access Action statement1 AND staement2 AND statement OR.

multiple http_access statements are OR'ed whereas elements of an access entries are AND'ed together; due to this reason the

http_access allow host1 morning evening

would have never worked as time morning and evening (morning AND evening) would never ever be TRUE and hence no action would have taken place.

4. Blocking sites
Squid can prevent the access to a particular site or to sites which contain a particular word. This can be implemented in the following way

acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex abc.com *()(*.com
http_access deny banned_sites
http_access allow allowed_clients

The same can also be used to prevent access to sites containing a particular word i.e. dummy , fake

acl allowed_clients src 192.168.0.1/255.255.255.0

```
acl banned_sites url_regex dummy fake
http_access deny banned_sites
http_access allow allowed_machines
```

It is not practical to list all the words list or sites names to whom the access is to be prevented; these can be listed out in the file (say banned.list in /etc directory) and ACL can pick up this information from this file and prevent the access to the banned sites.

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl banned_sites url_regex "/etc/banned.list"
http_access deny banned_sites
http_access allow allowed_clients
```

5. To optimize the use
   Squid can limit number the of connections from the client machine and this is possible through the maxconn element. To use this option, client_db feature should be enabled first.

   ```
   acl mynetwork 192.168.0.1/255.255.255.0
   acl numconn maxconn 5
   http_access deny mynetwork numconn
   ```

   Note:
   maxconn ACL uses less-than comparison. This ACL is matched when the number of connections is greater than the specified value. This is the main reason for which this ACL is not used with the http_access allow rule.
6. Caching the data
   Response of the request are cached immediately, this is quite good for static pages. There is no need to cache cgi-bin or Servlet and this can be prevented by using the no_cache ACL element.

   ```
   acl cache_prevent1 url_regex cgi-bin /?
   acl cache_prevent2 url_regex Servlet
   no_cache deny cache_prevent1
   no_cache deny cache_prevent2
   ```

7. Creating Your Own Error Messages
   It is possible to create your own error message with a deny rule and this is possible with the deny_info option. All the Squid error messages by default are placed in the /etc/squid/errors directory. The error directory can be configured with the error_directory option. You can even customize the existing error messages.

   ```
   acl allowed_clients src 192.168.0.1/255.255.255.0
   acl banned_sites url_regex abc.com *()(*.com
   http_access deny banned_sites
   deny_info ERR_BANNED_SITE banned_sites
   http_access allow allowed_clients
   ```

   In the above example, a special message will be displayed when ever users try to access the sites with above banned words.The file name in the option i.e.ERR_BANNED_SITE must exist in the

above error directory. This error message file should be in HTML format. The above listed out examples are just a few of the options, facilities and capabilities of ACL. One can read through the FAQ section at the Squid Home Page for more extensive usage and explanation of other ACL elements and access elements.

Log Files

All log files of Squid are contained in directory /var/log/squid; these contain cache log, access logs and store.log. File access.log maintains the information about the clients request, activity and maintains entry for each HTTP & ICP queries received by the proxy server, clients IP, request method, requested URL, etc.. The data of this file can be used to analyze the access information. Many programs like sarg, calamaris, Squid-Log-Analyzer are available which can analyze this data and generate reports (in HTML format). The reports can be generated in terms of users, IP numbers, site visited, etc..

The destination of these log files can also be changed by following options

```
cache_access_log      For access.log
cache_log             For cache.log
cache_store_log       For store.log (Store manager)
pid_filename          Squid process ID file name
```

Authentication Methods

Squid in the default configuration allows any user to have access without any authentication process. To authenticate the users i.e. to allow only valid users (from any machine in the network) to access the Internet, Squid provides for authentication process but via an external program, for this a valid username and password is required. This is achieved by using proxy_auth ACL and authenticate_program; which forces a user to verify the username and password before the access is given. Several authentication programs are available which Squid can use and these are

1. LDAP : Uses Linux Lightweight Directory Access Protocol
2. NCSA : Uses NCSA style username and password file
3. SMB : Uses SMB server like SAMBA or Windows NT
4. MSNT : Uses Windows NT authentication domain
5. PAM : Uses Linux Pluggable Authentication Modules
6. getpwam : Uses Linux password file.

One needs to specify the authentication program being used and this can be specified by using the authenticate_program option. Make sure that the authentication program being used for the purpose is installed and working.

The changes in the squid.conf file now should also reflect the same authenticate_program /usr/local/bin/pam_auth

```
acl pass proxy_auth REQUIRED
acl mynetwork src 192.168.0.1/255.255.255.0
http_access deny !mynetwork
http_access allow pass
http_access deny all
```

This uses the PAM authentication program and all users need to authenticate before accessing the Internet.

Options like authenticate_ttl and authenticate_ip_ttl can also be used to change the behavior of the authentication process i.e. revalidation of username and password.

References

This article just touches the tip of the Squid iceberg; for further reference visit the following Web sites

- Squid Home, www.squid-cache.org
- Squid Documentation Project, squid-docs.sourceforge.net
- visolve.com
- For Proxy Authentication, home.iae.nl/users/devet/squid/proxy_auth

---

2005-01-14, generated by lfparser_pdf version 2.51