Transparent Proxy with Squid mini−HOWTO

Table of Contents

Transparent Proxy with Squid mini−HOWTO

Daniel Kiracofe

v1.0, 01 April 2000

This document provides information on how to setup a transparent caching HTTP proxy server using only Linux and squid.

7.Further Resources

1.Introduction

1.1 Comments

Comments and general feedback on this mini HOWTO are welcome and can be directed to its author, Daniel Kiracofe, at drk@unxsoft.com.

1.2 Copyrights and Trademarks

1.3 #include <disclaimer.h>

No warranty, expressed or implied, etc, etc, etc...

2.Overview of Transparent Proxying

2.1 Motivation

In ``ordinary'' proxying, the client specifies the hostname and port number of a proxy in his web browsing software. The browser then makes requests to the proxy, and the proxy forwards them to the origin servers. This is all fine and good, but sometimes one of several situations arise. Either

- You want to force clients on your network to use the proxy, whether they want to or not.
- You want clients to use a proxy, but don't want them to know they're being proxied.
- You want clients to be proxied, but don't want to go to all the work of updating the settings in hundreds or thousands of web browsers.

This is where transparent proxying comes in. A web request can be intercepted by the proxy, transparently. That is, as far as the client software knows, it is talking to the origin server itself, when it is really the proxy server.

Cisco routers support transparent proxying. But, (surprisingly enough) Linux can act as a router, and can perform transparent proxying by redirecting TCP connections to local ports. However, we also need to make our web proxy aware of the affect of the redirection, so that it can make connections to the proper origin servers. There are two general ways this works:

The first is when your web proxy is not transparent proxy aware. You can use a nifty little daemon called transproxy that sits in front of your web proxy and takes care of all the messy details for you. transproxy was written by John Saunders, and is available from ftp://ftp.nlc.net .au/pub/linux/www/ or your local metalab mirror. transproxy will not be discussed further in this document.

A cleaner solution is to get a web proxy that is aware of transparent proxying itself. The one we are going to focus on here is squid. Squid is an Open Source caching proxy server for Unix systems. It is available from www.squid−cache.org

2.2 Scope of this document

This document will focus on squid version 2.3 and linux kernel version 2.2, the most current stable releases as of this writing (March 2000). It should also work with squids as early as 2.0 and the later 2.1 linux kernels. Should you need information about earlier releases, you may find some earlier documents at www.unxsoft.com.

If you want to use linux 2.3, you will have to use a thing called netfilter instead of ipchains. However, it is assumed that if you are running a development kernel, you can figure out netfilter on your own from the provided documentation. If not, you really shouldn't be running a development kernel (trust me on this). Once linux 2.4 is released, this document will be updated to cover netfilter.

3.Configuring the Kernel

First, we need to make sure all the proper options are set in your kernel. If you are using a stock kernel from your distribution, transparent proxying may or may not be enabled (IIRC, it is in RH 6.1, but don't quote me on that). If you are unsure, the best way to tell is to simply skip this section, and if the commands in the next section give you weird errors, it's probably because the kernel wasn't configured properly.

If your kernel is not configured for transparent proxying, you will need to recompile. Recompiling a kernel is a complex process (at least at first), and it is beyond the scope of this document. If you need help compiling a kernel, please see The Kernel HOWTO.

The options you need to set in your configuration are as follows (Note: none of these can be built as modules)

- Sysctl support
- TCP/IP networking
- IP: firewalling
- IP: always defragment
- IP: transparent proxy support
- /proc filesystem support

Once you have your new kernel up and running, you may need to enable IP forwarding. IP forwarding allows your computer to act as a router. Since this is not what the average user wants to do, it is off by default and must be explicitly enabled at run−time. However, your distribution might do this for you already. To check, do ``cat /proc/sys/net/ipv4/ip_forward''. If you see ``1'' you're good. Otherwise, do ``cat '1' > /proc/sys/net/ipv4/ip_forward''. You will then want to add that command to your appropriate bootup script in /etc/rc.d/.

4.Setting up squid

ow, we need to get squid up and running. Download the latest source tarball from www.squid−cache.org. Make sure you get a STABLE version, not a DEVEL version. The latest as of this writing was squid−2.3.STABLE1.tar.gz.

Now, untar and gunzip the archive (use ``tar −xzf <filename>''). Run the autoconfiguration script (``./configure''), compile (``make'') and then install (``make install'').

Now, we need to edit the default squid.conf file (installed to /usr/local/squid/ etc/squid.conf, unless you changed the defaults). The squid.conf file is heavily commented. In fact, some of the best documentation available for squid is in the squid.conf file. After you get it all up and running, you should go back and reread the whole thing. But for now, let's just get the minimum required. Find the following directives, uncomment them, and change them to the appropriate values:

- httpd_accel_host virtual
- httpd_accel_port 80
- httpd_accel_with_proxy on httpd_accel_uses_host_header on

Finally, look at the http_access directive. The default is usually ``http_access deny all''. This will prevent anyone from accessing squid. For now, you can change this to ``http_access allow all'', but once it is working, you will probably want to read the directions on ACLs (Access Control Lists), and setup the cache such that only people on your local network (or whatever) can access the cache. This may seem silly, but you should put some kind of restrictions on access to your cache. People behind filtering firewalls (such as porn filters, or filters in nations where speech is not very free) often ``hijack'' onto wide open proxies and eat up your bandwidth.

Initialize the cache directories with ``squid −z'' (if this is a not a new installation of squid, you should skip this step).

Now, run squid using the RunCache script in the /usr/local/squid/bin/ directory. If it works, you should be able to set your web browser's proxy settings to the IP of the box and port 3128 (unless you changed the default port number) and access squid as a normal proxy.

For additional help configuring, see the squid FAQ at www.squid−cache.org.

5.Setting up ipchains

ipchains should be installed with almost every recent distribution (anything based on kernel 2.2). However, should you not have ipchains, you can get it from ftp://ftp.rustcorp.com/ipchains/. ipchains is a very powerful tool, and we'll only scratch the surface here. For more information, please see http://www.rustcorp.com/linux/ipchains/HOWTO.html for the ipchains HOWTO.

To set up the rules, you will need to know two things, the IP address of the box (I'll use 192.168.1.1 as an example) and the port squid is running on (I'll use the default of 3128 as an example).

First, we need to allow packets destined for any actual webserver on this box through. We should setup both the loopback interface and the ethernet interface. You should not skip this step even if you no actual webserver on your box, as the absence of these rules can create infinite forwarding loops where the proxy tries to connect to itself. Use the following commands:

- ipchains −A input −p TCP −d 127.0.0.1/32 www −j ACCEPT
- ipchains −A input −p TCP −d 192.168.1.1/32 www −j ACCEPT

Now, the magic words for transparent proxying:

- ipchains −A input −p TCP −d any/0 www −j REDIRECT 3128

You will want to add the above commands to your appropriate bootup script under /etc/rc.d/.

6.Put it all together

If everything has gone well so far, go to another machine, change it's gateway to the IP of your new squid box, and surf away. To make sure that requests are really being forwarded through your proxy instead of straight to the origin server, check the log file /usr/local/squid/logs/acces s.log

7.Further Resources

Should you still need assistance, you may wish to check the squid FAQ or the squid mailing list at www.squid−cache.org. You may also e−mail me at drk@unxsoft.com, and I'll try to answer your questions if time permits (sometimes it does, but sometimes it doesn't)