SSH+SQUID HOWTO

Brian St. Pierre

<SSH+SQUID-HOWTO@bstpierre.org>

2004-07-05

Revision History

Revision 0.1	2004-07-05	BSS
First published draft		

This document describes how to set up a secure web proxy using SSH and Squid. Such a setup can, among other things, be used to circumvent web censorship.

This is an early draft. I'm actively seeking feedback. Send email to <SSH+SQUID-HOWTO@bstpierre.org>. Thanks in advance for your help.

Table of Contents

Introduction

Many organizations (e.g. public libraries, employers) and some countries (e.g. Iran, China) have put mandatory web censorship in place through the use of filters like Websense or by blocking off large portions of the Internet at the (national) network. One way of getting around this is by setting up a proxy server that is hosted outside the purview of the censoring organization, but is still accessible from computers on the censored network.

In order for any of this to work, you need:

- A computer that you control (the "server") with an Internet connection hosted outside the censored network.
- A static IP address is helpful, but not required.
- It is helpful to be running linux on the server, but not required.
- Sufficient administrative privileges on the censored computer (the "client") to be able to install software -- or you need to have an SSH client already installed on the client.

Unless specified otherwise, it is assumed that all commands below are run as root on the server.

Credits / Contributors

Many thanks to:

- Joe DiPerna for providing an outline of how to get started.

  Olivier Sessink for creating jailkit.

  The fine folks responsible for openssh and squid.

SSH

Install SSH on the Server

You need to have an SSH server installed on the server. If you are running Debian, this is "apt-get install ssh". If you are running something else, install the ssh package from your distribution or go to http://www.openssh.com/ and follow the installation instructions.

Start SSH

You can manually start SSHD by running "/etc/init.d/ssh start" (Debian), or "service sshd start" (Red Hat). However, this may not cause SSHD to start when the server is rebooted. You want SSHD to start every time the server boots. To do this, use "ksysv" (Debian) or "chkconfig" (Red Hat).

On Debian, you may need to remove the file /etc/ssh/ssh_not_to_be_started so that SSHD will start.

Test SSH

It is preferable to test from another machine if you have one handy. In a pinch you can test from the same machine. At a prompt, run "ssh USERNAME@SERVER-NAME-OR-IP-ADDRESS". You should be prompted for a password, and eventually get a shell prompt.

Squid

Squid is a caching web proxy, and is the guts of this setup. It is also the most complicated to configure.

Install Squid

Install Squid using "apt-get install squid" (Debian) or the appropriate command for your distribution. If all else fails, go to http://www.squid-cache.org/ and follow the download, installation, and configuration instructions there.

Configure Squid

Squid configuration is a bit hairy. I installed from a Debian package and accepted most of the defaults. My changes are outlined here:

1. In the ACL section, disable a bunch of ports that are not going to be used. I'm only enabling the main SSL and SSH ports, in addition to Gopher and FTP. Disable these ports by removing or commenting out the corresponding lines.

2. Still in the ACL section, define an acl for "our_networks" (this is in the default config I have) that is 127.0.0.1/32. This should be the only connection you allow -- essentially restricting access to the proxy to the local machine.

If you have problems configuring Squid, check the website above.

Start Squid

Starting Squid is eerily similar to starting SSHD see the section called “Start SSH”. The squid startup script is "/etc/init.d/squid".

You’ll need to restart Squid when/if you make changes to the configuration file.

Test Squid

From the server, fire up a browser. Configure the browser to use localhost:3128 as a proxy. Go to a few different websites. Check the log file (on Debian this is in /var/log/squid/access.log). You should see one (possibly several) entry for each web page you visited.

Celebrate. The hard part is over.

Jailkit & Users

> Jailkit is a set of utilities to limit user accounts to specific files using chroot() and or specific commands. Setting up a chroot shell, a shell limited to some specific command, or a daemon inside a chroot jail is a lot easier using these utilities.
>
> —From the website (http://olivier.sessink.nl/jailkit/)

Jailkit is highly recommended if you’re going to allow other users (especially people you’ve never met and/or don’t completely trust). Many security holes are only vulnerable to local users. You can reduce your exposure to security holes by limiting the programs that your users can access. For the purposes of setting up a secure web proxy, they really don’t need anything beyond a login shell.

Jailkit is recommended even if you’re just setting this up for yourself. In this case, you should create a separate "jailed" account for yourself. This limits your exposure in the case where the client computer is hijacked or otherwise compromised.

Install Jailkit

Go to the website referred to above, click the download link, and grab the latest source tarball. From a prompt, run the following commands. (If you are paranoid, run the first four commands as a non-root user.)

```
$ tar zxvf jailkit-VERSION.tar.gz
$ cd jailkit-VERSION
$ ./configure
$ make
$ make install
```

Prepare the jail

Follow these instructions at the jailkit website [http://olivier.sessink.nl/jailkit/howtos_chroot_shell.html].

Note that "basicshell" is the only required piece of initialization. You don't really need to give your users editors or anything -- they're just going to be forwarding data via SSH.

Run the "jk_check" command when you're done and put this command in your crontab as suggested by the instructions referenced above.

Create Additional User Account(s)

If more than one user will be using the proxy and you want to keep them separate, you can create an additional user and perform the steps in the section called "Prepare the jail" again.

Test Jailkit

From another machine, run:

```
$ ssh -L 3128:SERVER:3128 -l JAILED_USERNAME SERVER
```

You should get a login prompt. Try running "ls" or some other command at the prompt. You should get "command not found" errors -- which means the jail is working properly.

Configure the Client

Configure your Browser

Now launch a browser and change your proxy to localhost:3128. Detailed instructions for particular browsers are below.

Firefox 0.8

1. Tools->Options
2. Select the General tab.
3. Push the Connection Settings button.
4. Choose Manual Proxy Configuration.
5. In the HTTP box, type "localhost".
6. In the Port box next to that, type "8118".
7. Push OK to close Connection Settings.
8. Push OK to close Options.
9. Test the changes by going to http://www.google.com/