The Sysadmin's Daily Grind: Tor

SECRET SURFING

Some people don't mind leaving traces of their IP address wherever they go, others prefer to use a tool like The Onion Router.

BY CHARLY KÜHNAST

The Onion Router (Tor) [1] runs as a Socks 4 proxy and mangles incoming connections through a network of distributed, independent servers, thus removing any traces of the original data packets. The Onion is similar to Java Anonymous Proxy (JAP), a tool that supports anonymous web surfing. Your ex won't be able to evaluate the IP address of a blog entry to find out who added those unflattering comments about her. The IP address will be that of the last server in the Onion routing chain.

I downloaded Tor from [2], which has a collection of ready-made packages for a variety of Linux distributions, BSD derivates, MacOS, and Windows. I decided to build from scratch using the current tarballs, a quick process. I had to install the OpenSSL and Libevent libraries on my machine, but then it was standard procedure: *./configure && make && make install*. As I did not specify any preferences for the target directory, the binaries ended up in */usr/local/bin*, and a sample configuration file was placed in */usr/local/etc/tor/*.

You might like to set up a Tor user instead of running Tor with root privileges. Tor supports client and server mode operations. Client mode is useful if your computer has a private, non-routable IP address, for example 10.*x.x.x* and uses a NAT gateway for Internet connections. Client mode is also the right choice for computers with modems.

The configuration is quite simple. After launching, Tor listens on port 9050 for incoming Socks 4 connections. You can then point your browser at this port to route the connection via the Onion routing network. The documentation correctly points out that it is a good idea to use a combination of Tor and Privoxy for surfing. For more details, check out [3].

Socks with Nodes

If you have a public IP address and enough bandwidth, you will want to run Tor in server mode. The advantage of server mode is that your computer then becomes a node on the Onion routing network. The more nodes the network has, the more efficient it becomes, and the better the performance is. Nothing changes from the Socks application's point of view – the server is still a Socks 4 proxy.

You'll need to watch out for a few things if you are running Tor in server mode. Your system clock should be as precise as possible; you can sync with a timeserver to achieve this. You also need to add a nickname for the server to the excellently documented *torrc* configuration file, and you also need to say which port the server will listen to for incoming connections. A number of other settings make good sense, such as bandwidth limits or networks from which you want to allow or deny incoming connections.

The *exit policy* is also worth a read. This policy conceals Tor's mechanism for preventing connections to specific target ports. The program will not route email in this case, even if a mail client can use Socks proxies. This feature prevents spammers from misusing your Tor server. Have fun playing hide and seek with Tor. And don't let me catch you writing anonymous messages about my weight in my blog! ■

INFO

[1] Tor: *http://tor.eff.org*

[2] Download: *http://tor.eff.org/download.html*

[3] Combining Tor and Privoxy: *http://tor.eff.org/cvs/tor/doc/tor-doc.html#client-or-server*

THE AUTHOR

Charly Kühnast is a Unix System Manager at the data-center in Moers, near Germany's famous River Rhine. His tasks include ensuring firewall security and availability and taking care of the DMZ (demilitarized zone).