The Monthly GNU Column

BRAVE GNU WORLD

This column looks into projects and current affairs in the world of free software. This month we focus on better security with Firestarter, a tool that helps you set up a firewall. We also look at a distributed computing project for animation or visualization. **BY GEORG C. F. GREVE**

Firewalls are more important now than ever, so a free tool that helps users manage their firewalls is always welcome.

Firewalls

A firewall operates at the network connection level, analyzing, monitoring, and controlling the traffic that enters or leaves a computer. On Linux with kernel 2.4 or later, this is the domain of iptables. Network traffic is represented by IP data packets, and the kernel specifies what should happen to specific types of packets. Firewalls of this type are known as packet filters.

Firestarter [5] is a graphical firewall administration program by Tomas Junnonen. The program aims to make the job of setting up a Linux firewall more simple, convenient, and transparent.

When a user first launches the program, a firewall wizard takes the user through an initial setup. The wizard allows users to complete a basic configuration with the focus on distributed Internet connections for multiple computers using dynamic IP address assignments.

Figure 1: Firestarter displaying an overview of the current firewall status.

The default packet filter settings allow any outgoing connections and deny any external packets that attempt to access the computer, with the exception of packets that belong to existing connections. Firestarter has three tabs called *Status*, *Events*, and *Policy* for normal operations. The *Status* tab in Figure 1 not only shows you the current firewall status, but also gives you information on current connections, including the applications and services responsible for those connections. It is also possible to modify settings made using the wizard and to enable or disable the firewall. Firestarter also has a single-click detach feature that prevents any communication with the outside.

Access Monitoring

The *Event* tab displays blocked connection attempts as events, giving you details on the origin, service, and time (Figure 2). You can click to add a filtering rule to allow a service or a host – or to allow exactly this service from exactly this computer. Simple management is one of Firestarter's strongest points. Instead of wading through tables of port numbers to discover the services assigned to them, users can simple check the *Event* tab to allow access. And the *Policy* allows you to remove any unwanted rules.

Priorities

Firestarter also supports ICMP packet filtering. ICMP was originally designed for network diagnostics, but it is often misused for Denial-of-Service attacks. Additionally, the *Type of Service* feature allows you to prioritize services and thus optimize network traffic for throughput, reliability, or interactivity. One of Firestarter's neatest features is that it displays the firewall status in the panel and alerts you to specific events.

Tomas started work on Firestarter back in 2000 while he was waiting for a place at the university after completing national service. At the time, he noted that none of the available programs did exactly what he wanted it to do. In other words, Firestarter, like many other pro-

grams, started life with the famous hacker mantra: "That can't be too difficult."

Firestarter is a free, GPLed application that was written in C using the GTK+ toolkit. As the program was recently added to the Gnome CVS tree, the project's translation team has added localized versions, and errors are handled by the Gnome bugtracker.

One of the issues the team faced was the small but important differences between distributions. However, a team of volunteers, including Netfilter guru Paul Drain, brought the project to maturity, seeing the release of version 1.0 in November 2004, and providing binary packages for a number of distributions. For Tomas, Firestarter's major strengths are its ease of use and good supporting documentation – although he admits that, like many other free projects, Firestarter could benefit from more professional graphics.

Firestarter really does take the fear factor out of iptables configuration for normal users, although it does not currently support the full range of firewall configurations and may not cover some complex scenarios.

Distributed Processing

Let's move on now to part two of this month's column and look at distributed processing, more specifically, at the rendering manager DrQueue [6]. The program's author, Jorge Daza, first released this GPLed tool about a year ago. DrQueue distributes the task of rendering multiple individual images or animations over a pool of computers.

The interesting thing is that the computers can use different rendering tools. DrQueue supports both proprietary programs such as Maya, Mental Ray, and Blue Moon Rendering Tools (BMRT), and free software applications such as

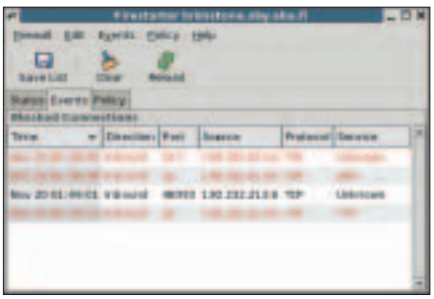

Figure 2: The overview has an entry with the connection details for each denied packet.

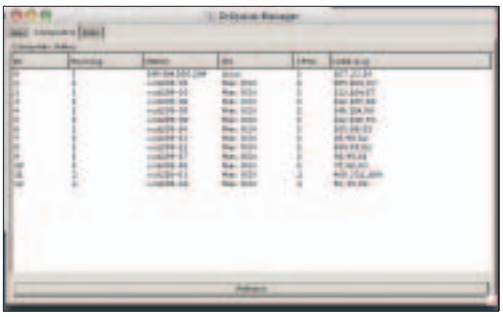

Figure 3: Dr Queue controls CPU capacity for rendering individual images within a digital animation. The GUI gives users an overview of the network nodes.

Blender [7] and Pixie [8]. You can add more rendering tools provided they have a default profile that matches DrQueue's requirements. A simple shell script handles task distribution.

Easily Ported

The program itself was mainly written in C, with a few lines of C++ and TCSH for scripting thrown in for good measure. DrQueue uses Scons [9] to support configuration and compilation, and this means you can run the program on Linux, FreeBSD, Irix, and Mac OS X.

The resource manager has three major elements known as the master, slave, and drqman. The master assumes the server role, handling task distribution and coordinating the results, whereas the slave runs on the individual nodes to handle processing as stipulated by the master program. drqman provides a convenient GUI for users based on the GTK+ toolkit.

Central Data Repository

The program collects results and logfiles on a Network File System (NFS) export, which is mounted by all the computers in the group. As NFS is a fairly ancient protocol that does not provide a lot in the line of security or cryptography, and as the queue manager does not add these capabilities, it makes sense to isolate this kind of setup from the rest of your network or hide it behind a securely configured firewall. It might also be a good idea to make sure the computers involved can only exchange data within the group; the Firestarter tool we just looked at would be the ideal tool to handle this.

One of DrQueue's major strengths is its ability to priorize tasks and specify dependencies between tasks. Additionally, the program groups slaves in pools and allows granular control of the number of CPUs assigned to a job. DrQueue also assigns processing work based on the operating system, the system load, and several additional parameters.

The master program maintains status information concerning crashes or system resets, allowing processing to pick up from where it left off after an interruption. Users can access these parameters via the GUI while processing is in progress and tell the program to reprocess individual frames, remove computers from the pool, or add computers to the pool.

As the current version already has all the features that Jorge Daza planned for version 1.0, there is nothing to prevent the release from going ahead. However, Jorge is looking to improve usability and the help feature. The author also intends to simplify the install and add binary packages for major distributions.

Documentation

Documentation is a major prerequisite to usability. Unfortunately, the existing documentation is incomplete and not up to date. Jorge would appreciate your feedback. If you are interested and have the required technical skills, please check out the project and help where you can. ■

INFO

[1] Send ideas, comments, and questions to Brave GNU World: *column@brave-gnu-world.org*

[2] GNU project homepage: *http://www.gnu.org/*

[3] Georg's Brave GNU World homepage: *http://brave-gnu-world.org*

[4] "We run GNU" initiative: *http://www.gnu.org/brave-gnu-world/rungnu/rungnu.en.html*

[5] Firestarter homepage: *http://www.fs-security.com*

[6] DrQueue homepage: *http://www.drqueue.org*

[7] Blender homepage: *http://www.blender.org*

[8] Pixie homepage: *http://pixie.sf.net*

[9] Scons homepage: *http://www.scons.org/*