


GUI-based firewall configuration with KMyFirewall

VISIBLE SECURITY



Linux has a fantastic selection of firewalls for securing stand-alone computers or whole networks. Although you can use IPTables to set up a firewall, the configuration is often the most difficult step. KMyFirewall offers a powerful, user-friendly, GUI-based approach. **BY ERIK BÄRWALDT**

In our age of the global Internet, with computers permanently exposed to danger from hackers, it is more important than ever to protect your own systems against attack. The firewall has become a first line of defense for the network. Luckily for Linux users, your favorite free operating system has an integrated firewall, and more developers are starting to provide a usable firewall configuration as part of the minimal installation of their Linux systems. This removes the need for users to invest heavily in security – in contrast to what users of other, more vulnerable systems face.

Linux has an enormous range of firewall systems, from application-oriented

firewalls such as Firestarter or FW-Builder, which mainly rely on the default security systems in Linux, IPTables/Netfilter, through turnkey firewall distributions that boot from CD or DVD, such as IPCop, which typically integrate some kind of server functionality, to expensive commercial solutions such as Check-Point, InJoy, or gateProtect.

Commercial firewalls typically offer value-added functionality, such as VPN tunneling capabilities for secure networking, or QoS (Quality of Service), in other words, bandwidth control and load distribution, both of which are useful in large LANs with low bandwidth DSL or even ISDN-based Internet connectivity.

If you look at the firewall configuration, commercial programs do not typically offer more functionality than open source solutions. Thanks to the IPTables/Netfilter firewall, which was first added to the Linux mainstream in kernel 2.4, open source programs also have a powerful base system that provides both basic packet filtering and stateful inspection, and thus the ability to analyze, and if needed, block data streams.

Background Theory

If you are interested in security, and mean to take the topic seriously, there are a few basic facts that you need to know, unless you want your firewall to

have more holes than Swiss cheese. The underlying protocols covered by firewall configuration are TCP, UDP, and ICMP.

To keep things simple, let's just say that the TCP protocol mainly acts as the control protocol for IP-based communications. Two computers that use TCP/IP to talk to one another rely on the three-way-handshake to establish the connection, and negotiate when they need to send acknowledgments. If an acknowledgment goes astray, the data transmitted since the last acknowledgment was received are retransmitted to ensure the integrity of the data stream.

TCP uses ports to allow multiple data streams to reach different target applications within a single connection. Every application has a default port server-side, and the applications uses this port to send data streams to targets; client-side ports are configurable.

The UDP protocol resides in Layer 4 of the ISO/OSI reference model, just like TCP. However, in contrast to TCP, UDP is connectionless. In other words, the data transmission is not verified. The lack of verification makes UDP faster than TCP, of course.

In contrast to these two protocols, ICMP resides in Layer 3 of the ISO/OSI reference model and is used to transmit error and diagnostics messages. Ping is probably the best-known ICMP-based application and is used to check the

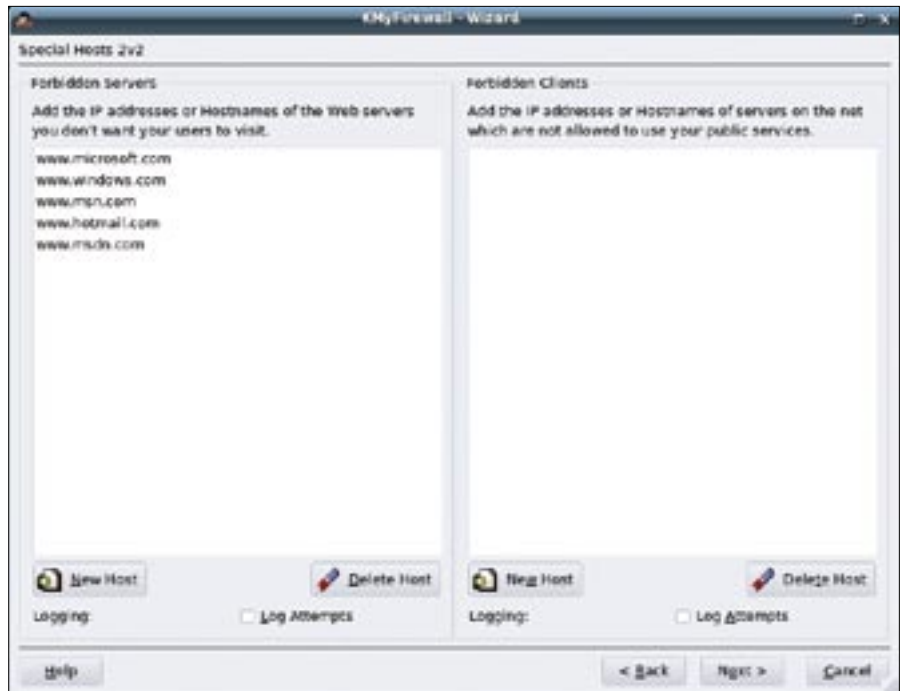


Figure 2: Blocking undesirable servers.

availability of a host on the Internet or a local network.

IPTables/Netfilter in Linux

IPTables/Netfilter replaced the previous Linux firewall systems, IPChains and – many years ago – IPFWadm in kernel version 2.4.x. This said, the command syntax in IPTables is more or less identical to the syntax used by IPChains, which is useful for administrators with

a long service record. As the name suggests, the IPTables firewall is configured as a clear text table.

Thus, the rules and chains in a text file define what to do with data packets. A ruleset is applied to each data packet. In addition to simple packet filtering, IPTables/Netfilter can also handle Network Address Translation (NAT). For IPTables/Netfilter to work, you need to configure the kernel appropriately by enabling the firewall when you build the kernel.

This said, more or less any recent distribution will have IPTables support built into the kernel, and you just need to enable your IPTables ruleset table in the appropriate runlevel to enable the table whenever you boot the machine.

However, there are always two sides to an equation, and no matter how neat it may sound to have the ability to configure rules in a table, you have to consider how cluttered the table can become and how easily this can lead to a configuration that simply does not work.

What's more, many users who have never experienced anything but point & click-style operating systems are afraid of having to do something this serious at the command line.

GUI-Based Firewall Management: KMyFirewall

The KDE window manager now has GUI-based tool, which is suitable for



Figure 1: The wizard helps you set up KmyFirewall.

firewall configuration in large-scale networks, KMyFirewall. KMyFirewall is based on the IPTables underpinnings, and offers the same functionality.

This said, KMyFirewall is not typically included in today's popular distributions. You can download the source packages from [1], and then go on to build and install it yourself by using the following steps:

```
root# ./configure --prefix=/
/KDE-Install-Path/
--with-qt-dir=/
/QT-Install-Path/
root# make
root# make install
```

The firewall requires the Qt3 libraries, as well as *automake* and *autoconf*; you might want to make sure that these packages are installed to avoid dependency issues. The install drops a *KMyFirewall* entry into the *System* menu, giving users the option of pointing and clicking to start the configuration.

After launching the firewall, and entering the root password, you will see a dialog where you can opt to use an assistant to create a new ruleset, or you can load a template for customization. You can also load a customized table for modification, or delete an existing ruleset to start from scratch.

After clicking *Wizard* to create a new table for a single workstation, the graphical wizard steps you through the ruleset-creating process. Filter rules are grouped logically by incoming and outgoing packet, or forwarded data streams if needed, and the wizard provides some basic information concerning the proto-

cols and their security issues at the same time (see Figure 1).

Users have the very important option of enabling logging at every stage, and logfiles will definitely help you investigate unclear situations. Granular controls for access to Internet services is another important feature. For example, you can deny workstations access to Internet-based FTP servers with a single click. KMyFirewall also gives you the ability to define malicious, trusted and forbidden servers, to which access is restricted or simply denied (Figure 2).

After completing the filter and function definition phase, you can check the text view of the table you just created. This check will be difficult for newcomers or if you are migrating from another system, as understanding the IPTables shell script assumes some knowledge of bash programming on Linux.

Variables as well as *if* and *case* constructs make the table hard to interpret or even understand for newcomers. If this is how you feel when you look at the table, you will probably prefer to check the ruleset in the main KMyFirewall window.

You will notice that the rules are displayed grouped by chain on the right-hand side of the window. Clicking the plus sign expands a chain and also displays the filter restrictions in place within the chain in a neat overview with green radio buttons.

As rulesets are always processed top-down, you can move a chain up or down. You can also use the editing function to modify individual rules to enable stateful inspection, for example, or to define thresholds, and thus prevent DoS

attacks (Denial of Service) (Figure 3).

After completing the firewall rule definition and saving your work, click *Run Firewall* to switch the firewall on. If you would prefer to enable the firewall automatically at system boot time, click the *Install Firewall* button and confirm the security prompt to add the process to the appropriate runlevel scripts.

Of course, you can launch KMyFirewall manually to modify rulesets whenever the need arises, for example, to shut down a security hole. And KMyFirewall gives experienced administrators the ability to protect larger or more complex networks in edit mode by manually entering new chains or rules, which include Network Address Translation (NAT) capabilities.

Conclusions

KMyFirewall is an extremely flexible tool for securing anything from a stand-alone workstation to a complete network. Its developers have been successful in hiding the complexity of security issues behind an easy-to-use GUI, but without also compromising any of the functionality of the powerful, underlying IPTables/Netfilter system.

The program is suitable both for more ambitious home users and for network administrators who run KDE on their server systems and who prefer a neat GUI to the possible risks of text-based IPTables rulesets.

This said, you will need to have at least some basic knowledge of networking technology, especially of the ISO/OSI reference model and of the major protocols within the model including their possible vulnerabilities.

As a word of warning, as a security-conscious user – no matter whether you are protecting your workstation at home or a server on an enterprise network – you should pen test your installation with tools such as Hping, Nmap, or Nessus to make sure that the firewall is working and can take the kind of punishment to be expected in the Wild. After all, it is infinitely preferable to find a security hole yourself rather than wait until an attacker does it for you. ■

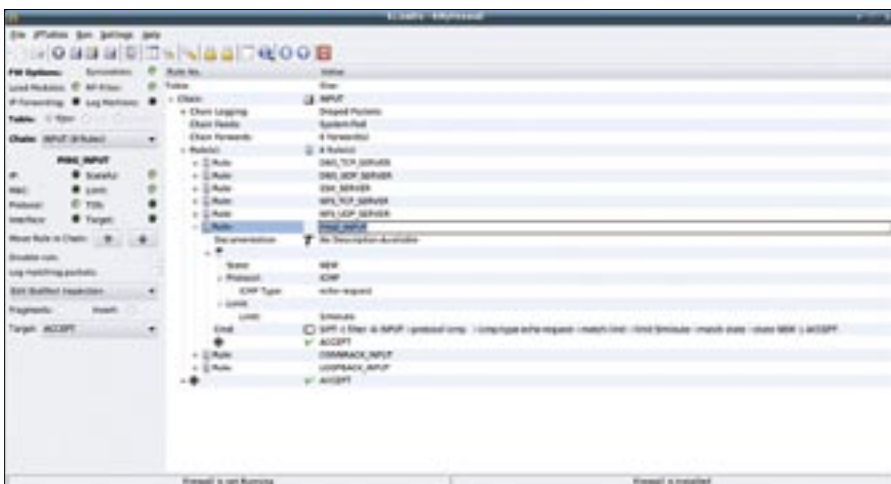


Figure 3: A clear overview of the rulesets.

INFO

[1] kmyfirewall homepage:
<http://www.kmyfirewall.org>