



Firestarter User's Manual

2005-01-07

<http://www.fs-security.com>

Index

- 1 Firestarter in a nutshell
- 2 A quick tutorial
- 3 Installation
- 4 Interface and basic usage
- 5 The firewall wizard
- 6 The status page
- 7 The events page
- 8 The policy page
- 9 The preferences
- 10 Working with policy
- 11 Internet connection sharing
- 12 Configuring the DHCP server
- 13 Advanced topics
- 14 Persistence of the firewall
- 15 Kernel requirements
- 16 Virtual Private Networking
- 17 Frequently asked questions
- 18 Developing Firestarter
- 19 Getting Firestarter from CVS
- 20 The Firestarter license

1. Introduction

Firestarter is an Open Source visual firewall program. The software aims to combine ease of use with powerful features, therefore serving both Linux desktop users and system administrators.

We strongly believe that your job is to make the high level security policy decisions and ours is to take care of the underlying details. This is a departure from your typical Linux firewall, which has traditionally required arcane implementation specific knowledge.

Why you need a firewall

A firewall does not guarantee security but it is in most environments the first line of defense against network based attacks.

You can use Firestarter on your...

... *desktop or laptop*. Our philosophy of simplicity has made Firestarter the most widely used Linux desktop firewall software available today.

... *server*. Firestarter can be installed onto individual servers and managed graphically over SSH or using the shell.

... *gateway or dedicated firewall*. Firestarter will set up Internet connection sharing for you with a minimum of fuss. Want DHCP for the clients? Sure you *could* configure it yourself, but we know you never get around to doing it, with Firestarter it only takes one click.

Firestarter features

- Open Source software, available free of charge
- User friendly, easy to use, graphical interface
- A wizard walks you through setting up your firewall on your first time
- Suitable for use on desktops, servers and gateways
- Real-time firewall event monitor shows intrusion attempts as they happen
- Enables Internet connection sharing, optionally with DHCP service for the clients
- Allows you to define both inbound and outbound access policy
- Open or stealth ports, shaping your firewalling with just a few mouse clicks
- Enable port forwarding for your local network in just seconds
- Option to whitelist or blacklist traffic
- Real time firewall events view
- View active network connections, including any traffic routed through the firewall
- Advanced Linux kernel tuning features provide protection from flooding, broadcasting and spoofing
- Support for tuning ICMP parameters to stop Denial of Service (DoS) attacks
- Support for tuning ToS parameters to improve services for connected client computers
- Ability to hook up user defined scripts or rulesets before or after firewall activation
- Supports Linux Kernels 2.4 and 2.6
- Translations available for many languages (38 languages as of November 2004)

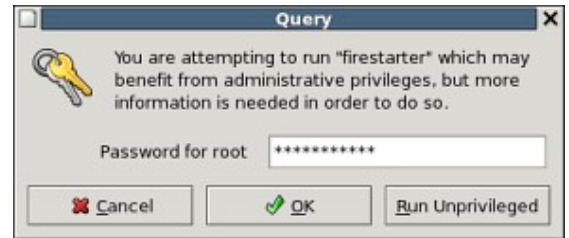


2. A quick tutorial

Starting Firestarter

After downloading and installing Firestarter, you will find the Firestarter icon in your desktop's programs menu. For example, in Fedora Core the Firestarter icon is located in the *System tools* menu. Alternatively you can run the program by simply executing "firestarter" from either a command line or from the *Run Application...* dialog (accessed by pressing Alt-F2).

Unless you are already logged in as root, you will be prompted for your root user password when starting Firestarter as a regular user.



Running Firestarter for the first time

Since you are running Firestarter for the first time, a wizard is launched. Following the welcome screen, you will be asked to select your network device from a list of detected choices for your machine. In case you have multiple devices, select the one that provides your Internet connection, otherwise you can use the default supplied.

In case your machine has multiple devices and can act as a gateway for your network, you will next have the option of sharing your Internet connection among all the computers on your local network. Again, simply select the local network connected device from the list of detected devices. If you wish for the clients to acquire their network settings automatically, simply check the option to *Enable DHCP for local network*.

Having completed the wizard, click the save button on page final page. The firewall is now ready and running, and your machine has an added layer of security. Firestarter now works in its default mode, which is a restrictive policy for incoming traffic and a permissive stance towards outgoing connections. This means you are fully protected against connection attempts from the outside, but are still able to browse the web, read your email, etc. as normal. There is no need to further configure Firestarter if you are satisfied with these defaults.

Read more about the wizard.

Trying out the Firestarter interface

Let's take a quick look at some of the features of the program itself. The application is divided into three pages, accessed through a tabbed notebook interface. These pages are *Status*, giving you an fast overview of state the firewall, *Events*, where blocked intrusion attempts and the firewall history is shown, and *Policy*, where you alter the behavior of the firewall by creating security policy.

From the Status page where you start out you can further access the preferences where you can change your network settings, as well as enable advanced options such as ICMP or ToS filtering. For now, let's take a look at the Events page.

Reacting to events

On the events page you will see all connections that the firewall has terminated since you started the program. By pressing the reload button you can also import all the previous events as recorded in the system log. This is really the core of the Firestarter program. Firestarter starts out in a restrictive mode, providing complete protection against incoming intrusions. That means that if you are running a legitimate service on your machine, for example a web server or SSH, connections to these services will also be stopped and recorded here at first.

Traditional firewalls will have you scrambling for the settings and configuration files at this point. However, when you see a connection attempt that you want to authorize, you simply right-click the entry in Firestarter and select "Allow inbound service for everyone". If you want to give access to the machine that is attempting the connection, but without even letting anyone else know that you're running the service in question, select "Allow inbound service for source". This is known as *stealth* and can be a very powerful tool.

Creating policy



The previous example of enabling the service could also have been accomplished from the Policy page. However, it is not just a gimmick, in reality you will want to create policy from events often for maximum security. By opening services to select machines only after the connection attempt, as shown above, you effectively minimize your exposure on the net. It's also very convenient.

Let's take a look at a legitimate reason to resort to the Policy page. Say Firestarter is running on your gateway, doing Internet connection sharing for your local network. On your local network you have a desktop, on which you wish to use the BitTorrent application. In the BitTorrent manual it tells you to "forward ports 6881-6889 from your firewall". With Firestarter this kind of setup is a piece of cake. Select the Policy page, right click on the list marked *Forward service* and select *Add rule*. You will be presented with a dialog for creating a new policy rule. Select BitTorrent from the service drop-down, fill in the IP of the client and you're done. Click the *Apply Policy* button to apply the changes.

Of course, that's only scratching the surface of what the Policy page can do. Another powerful feature is the ability to restrict outgoing traffic. For more information, refer the section on working with outbound policy.

Quitting the program

A frequently asked is question is, what happens when you quit the program. The answer is that the firewall will keep functioning. If you are running Firestarter as a system service, which is automatically set up for you when installing Firestarter from a binary package, the firewall is in many cases even running before you start the program.

3. Installation

Firestarter is packaged for many of the leading Linux distributions. Using a pre-compiled package ensures that the program will integrate properly with your distribution of choice. For platforms for which a binary package does not yet exist and for experienced users, Firestarter can also be compiled from source.

Installing in Fedora Core, Red Hat Linux, SuSE or Mandrake

Firestarter is conveniently available in RPM package format for RPM enabled Linux distributions like, Fedora Core, SuSE and Mandrake.

Once you have downloaded the Firestarter RPM specific to your distribution, open a terminal and change to the directory where you downloaded the RPM to. Type the following commands as shown in bold to install the package:

```
[bash]$ su
Password: [Type your root password and hit enter]
[bash]$ rpm -Uvh firestarter*rpm
Preparing...
...
```

Barring any unresolved dependencies or other problems, Firestarter should now be installed. Alternatively you can use a graphical package manager by double clicking the RPM file in your file manager.

Installing in Debian and Ubuntu

Firestarter is maintained in Debian and can be downloaded and installed using the *apt-get* tool by simply typing "apt-get install firestarter".

Ubuntu users can install Firestarter by enabling the "universe" repository in the */etc/apt/sources.list* file or in synaptic under Settings->Repositories. Having enabled the repository, the procedure is the same as in Debian.

Installing in Gentoo

Firestarter is fully supported in the Gentoo distribution by the *Portage* system. Simply run "emerge firestarter" to install the program.

Compiling and installing from source

Start by downloading the *tar.gz* version of Firestarter. Unpack the tarball and move into the newly created directory:

```
[bash]$ tar -zxvf firestarter*tar.gz
...
[bash]$ cd firestarter
```

Run the *configure* script. There is no need to give any parameters to the script, but we recommend you at least specify the *sysconfdir* variable, which determines the directory the firewall configuration will be written to. For a full list of options, see *./configure --help*.

```
[bash]$ ./configure --sysconfdir=/etc
checking for a BSD compatible install... /usr/bin/install -c
...
```

By default Firestarter will be installed into the */usr/local* tree when compiling from source, you can override this by setting the *prefix* option.

If the configure stage completed without problem you should now be able to compile and install the program:

```
[bash]$ make
...
[bash]$ su
Password: [Type your root password and hit enter]
[bash]$ make install
```

...

The *make install* stage is optional. You can also run Firestarter directly from the *src* subdirectory of the build tree if you want. In that case you must however first issue "*make install-data-local*" in the build directory. This will install the GConf configuration schema, Firestarter will not run without it.

Installing a Firestarter init script

When you install Firestarter from a package the program is automatically registered to run as a system service. This means the firewall is also running even if the graphical program is not. If you compile Firestarter from source and want this same functionality, you will have to install a system init script for your distribution.

In the firestarter tarball you will find *<distribution-name>.init* files. These are service startup scripts tailored to specific distributions, although you can likely use one even if it doesn't exactly match your distribution with a bit of editing.

To install the service, copy the init file to */etc/init.d/* and rename it to *firestarter.init*. After this you must tell the system to use the new script, exactly how this is done varies between distributions. If your distribution has the *chkconfig* tool available, simply run "*chkconfig firestarter reset*" and the service will be registered.

For more information about the Firestarter system service, refer to the section on firewall persistence.

4. Interface and basic usage

The main components of the Firestarter interface are:

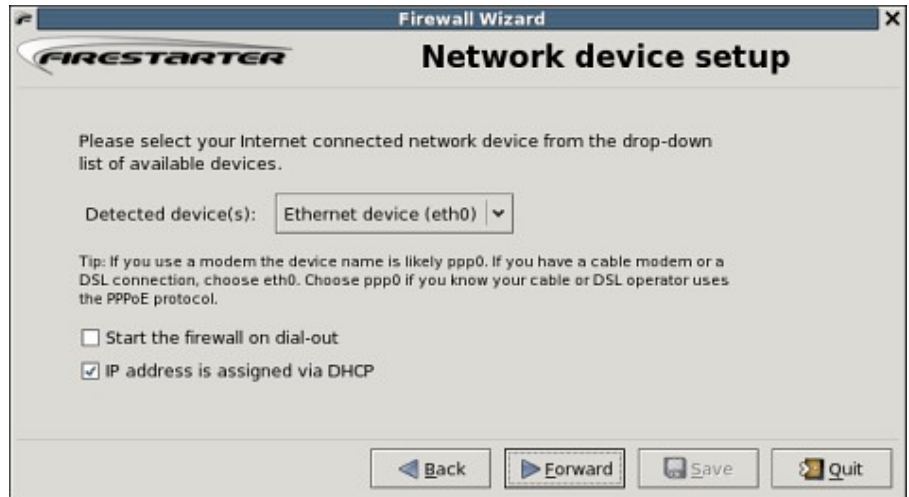
- The firewall wizard
The wizard guides you through configuring the application the first time you run Firestarter.
- The status page
This page in the main interface gives you a quick overview of the state of the firewall as well as allowing you start and shut it down.
- The events page
The second page in the main interface, the events page contains the intrusion attempt history of the firewall.
- The policy page
The final page in the main interface, the policy page is where you review your access policy. The policy alone determines what is allowed through the firewall.
- The preferences
The program preferences control many aspects of the interface, as well as giving you the option to enable some additional filtering functions of the firewall.

5. The firewall wizard

The Firestarter firewall wizard is automatically launched when you start the program the first time. If you want to return to the wizard at a later time, it is also accessible from the Firewall menu in the main interface. All of the choices made in the wizard can however also be changed through the preferences.

Network device setup

This page of the wizard is for configuring the primary network device; that is, your Internet connected network adapter.



The wizard automatically discovers all devices that are currently present in your machine. Generally, you will use either a pppxx or ethxx device, unless you have some special hardware. ppp is usually associated with a dial-up device while eth is the norm for most broadband connected machines.

Some cable modem users might have to select ppp0 as their device, even if there is also the choice of selecting eth0. This is because of the PPPoE protocol used by their ISP. If you see a ppp device in the list, and you do not have a modem, you probably must select it as your network device.

The following two options are available:

- *Start the Firewall on dial-out*
If you have a dialup (PPP) device or a VPN interface where the connection will be down occasionally, you should select this option. When enabled, Firestarter will try to reload the firewall as soon as the connection is established.
- *IP address is assigned via DHCP*
If your network settings such as the IP address of your computer is distributed via DHCP, you should enable this option. If you are connecting to the Internet using a cable, DSL or direct Ethernet connection you should almost always select this option. It is perfectly safe to select this option even if there is no DHCP server on your network. With this option enabled, Firestarter will reload the firewall when your network settings change.

Internet connection sharing setup

Internet connection sharing allows several machines to access the Internet through a single network connection. This is done using NAT. To the outside world the group of machines will look like a single machine with a single IP address.



For NAT to work you need two or more network devices in your machine. If you only have one device this page will not show in the wizard. To enable NAT, simply select a device from the drop down list of autodetected devices. You must select a

device other than the one you selected on the previous page.

For an in-depth look at the subject, as well as how to configure the DHCP service, read our guide to Internet connection sharing. Generally however, both NAT and DHCP will work out of the box simply by enabling them in the wizard, without the need to configure anything.

Ready to start the firewall

At the final page you have to option to either discard your changes or accept and save your choices. As soon as you click save, the firewall is started.

At this point Firestarter will be working in its default secure mode and there is no immediate need to further configure anything. The default mode implements a restrictive policy for incoming traffic and a permissive stance towards outgoing connections. For more information about the default mode of operation and how to change it, refer to the section on creating policy.

6. The status page

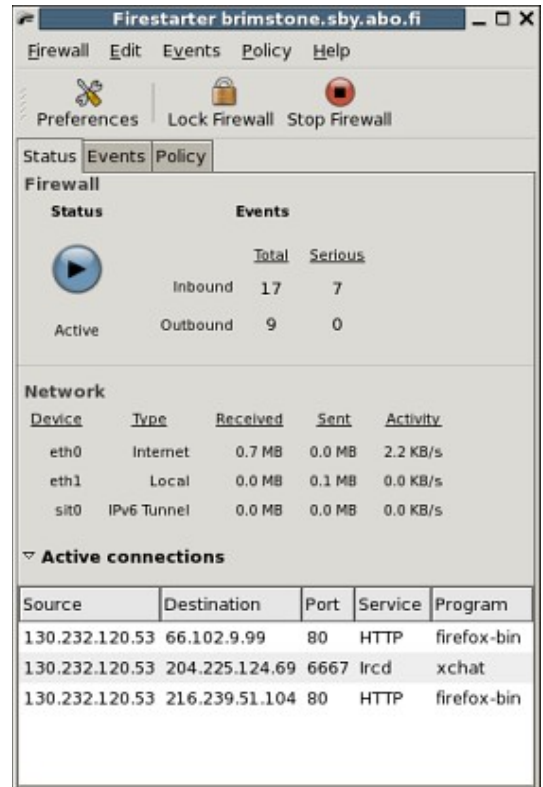
The status page is the first page you see when you start up Firestarter. This page gives you an overview of the firewall, as well as allowing you to change the state of the firewall.

The state is changed through the buttons on the toolbar at the top of the page. There are three states the firewall can be in:

- **Active**
The firewall is enabled and working.
- **Disabled**
The firewall has been stopped. In this state the security policy is not being enforced, all connections are accepted.
- **Locked**
The firewall is in a state of lock-down. Nothing is allowed through the firewall, neither in nor out.

A button on the toolbar of the status page allows you to access the program preferences.

The status page is further divided into three separate sections.



Firewall status

The firewall status section shows you at a glance the state the firewall is in. It does this through the icons depicted to the right.

A count is also kept of the number of firewall events, meaning blocked connection attempts, since program startup. The events are separated based on the direction of the attempted connection. If the connection originated from the firewall host or from a client on the LAN, it is classified as outbound, otherwise inbound. An event is further marked and counted as critical if Firestarter thinks it is a genuine threat and you should pay closer attention to it.

For more information about the various types of events, please see the section on events.



Network status

The network section of the status page gives you an overview of the network resource usage. It contains a list of all the network devices in the firewall host, as well as some metrics for each device.

The following device metrics are gathered:

- **Device**
The device name as reported by the operating system.
- **Type**
The role of the device in the firewall, or the generic device type if Firestarter is not using the device in question.
- **Received**
The amount of incoming traffic received through the device, in megabytes.
- **Sent**
The amount of outgoing traffic sent through the device, in megabytes.
- **Activity**
The current network bandwidth usage of the device.

Active connections

Active connections is a view into the firewall engine itself. It lists every established connection the firewall is tracking at any given moment. These connections include incoming traffic to the firewall as well as outgoing connections and the programs they are associated with. Furthermore, all the traffic that is being routed through the firewall, in case Internet connection sharing is enabled, is also tracked.

The following data points are recorded for each tracked connection:

- Source
The host that established the connection.
- Destination
The target host of the connection.
- Port
The network port the connection is using at the target host.
- Service
The name of the network service associated with the port in question.
- Program
The name of the program that created the program. This information is only available for connections local to the firewall host.

The entries in the active connections list are color coded as follows:

- Black
A currently active connection
- Gray
A terminated connection. Terminated connections are removed from the list after 10 seconds.

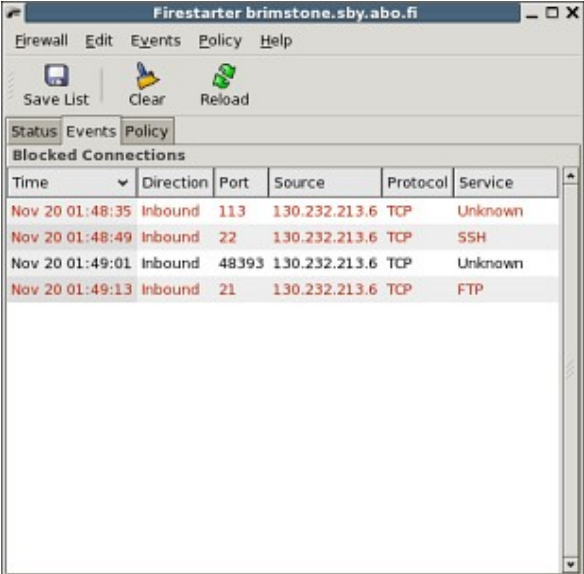
Using the context-sensitive menu associated with the entries, accessed through the right mouse button, there is the option to resolve the hostnames of the source and destination of a connection.

7. The events page

The events page shows the history of connections blocked by the firewall. As such, there is no need to act upon the entries. There are however many actions you can apply to the entries of denied connections, these then affect how the firewall will treat similar traffic in the future.

Firestarter color codes the entries according to how much attention you should pay them:

- **Black**
A regular connection attempt to a random port that was blocked by the firewall. Usually you do not need to worry about these.
- **Red**
A possible attempt to access a non-public service. Firestarter suggests you should pay special attention to these entries. Note that this does not have to mean someone is attempting an intrusion, only you can be the judge of that.
- **Gray**
Connections classified as harmless. This class consists mostly of broadcast traffic.



The screenshot shows the Firestarter application window titled "Firestarter brimstone.sby.abo.fi". The window has a menu bar with "Firewall", "Edit", "Events", "Policy", and "Help". Below the menu bar is a toolbar with "Save List", "Clear", and "Reload" buttons. The main area is divided into three tabs: "Status", "Events", and "Policy". The "Events" tab is active, showing a table of blocked connections. The table has columns for "Time", "Direction", "Port", "Source", "Protocol", and "Service". The data rows are as follows:

Time	Direction	Port	Source	Protocol	Service
Nov 20 01:48:35	Inbound	113	130.232.213.6	TCP	Unknown
Nov 20 01:48:49	Inbound	22	130.232.213.6	TCP	SSH
Nov 20 01:49:01	Inbound	48393	130.232.213.6	TCP	Unknown
Nov 20 01:49:13	Inbound	21	130.232.213.6	TCP	FTP

The event list

A number of data is gathered for each blocked connection. By default only a subset of these are shown, you can add more through the *Events->Show Column* menu.

The data columns available are:

- **Time**
The exact time the event occurred.
- **Direction**
The firewall classifies each connection as either inbound or outbound and this column shows you that class. See understanding policy for a detailed explanation of the directions.
- **In**
The network device the connection came in through.
- **Out**
The network device the connection was routed out through, if applicable.
- **Port**
The destination port of the connection.
- **Source**
The IP or hostname of the host that originated the connection.
- **Destination**
The IP or hostname of the target host of the connection.
- **Length**
The length of the blocked packet.
- **ToS**
The Type of Service parameters that were set on the packet.
- **Protocol**
The network protocol the connection used.
- **Service**
The name of the service the connection was trying to access.

By right-clicking on an event entry and selecting *Lookup Hostnames* from the context menu, the source and destination IP addresses can be converted into human readable hostnames.

The currently loaded list can be saved to disk in a human readable format through the save button on the toolbar.

The clear button will erase the currently visible events, while the reload button will load the entire firewall history from disk. Most Linux distributions will eventually rotate the logs in order to keep them from growing indefinitely, there is no need to purge the log manually. If the reloading of the events history is taking a very long time, it is possible to cancel the operation from the toolbar.

Acting on the event entries

Each entry in the list of blocked connections has a context sensitive menu associated with it, accessed by the right mouse button. These actions change how the firewall will treat a similar connection the next time it sees it. The actions act as shortcuts for manipulating the policy system, what they really do is create new rules on the policy page. Depending on the

type of event entry you are invoking the context menu on, you are presented with different actions.

The actions for the *inbound* entries are:

- **Allow Connections from Source**
This action gives the source of the connection permission to make any connection it wants. This is equivalent to trusting the source blindly and should be carefully used.
- **Allow Inbound Service for Everyone**
This action allows everyone to access the service the connection was previously blocked from.
- **Allow Inbound Service for Source**
This action allows only this specific source to access the service in question. This is known as *stealthing*, no other host except the source will be aware that the service even exists.

The actions for the *outbound* entries are:

- **Allow Connections to Destination**
By calling this action, you give everyone the permission to access the selected destination. This is equivalent to whitelisting the destination on the policy page.
- **Allow Outbound Service for Everyone**
This action allows everyone to make outgoing connections to the service in question, effectively whitelisting it.
- **Allow Outbound Service for Source**
This action gives this specific source outbound access to the service in question. Other clients on your local network will not be able to use the service.

Outbound entries generally only show up when you have restrictive outbound policy.

Filtering the events

Besides the policy actions available in the context menu for the event entries, there are also two options that help to manage the list itself. These are *Disable Events from Source* and *Disable Events on Port*.

As the names says, these options disable events from specific sources or on selected ports respectively. Selecting one of these options will therefore keep events matching the criteria of the list. Note that this has no effect on whether the connection is in fact blocked or not.

The list of currently filtered sources and ports can be found in the preferences.

8. The policy page

For a general explanation of the policy system, see working with policy. This section covers how to use the policy page of the Firestarter interface in practice.

The policy page is divided into two parts, the inbound traffic policy and the outbound traffic policy. To switch from one view to the other, use the drop down list at the top of the page.

Each view consists of three lists, called rule groups. A policy rule is simply an entry in one of the lists. To add a new rule, select the group you want to add it to by clicking the corresponding list. Then either click the *Add Rule* button on the toolbar or use the context sensitive menu by right-clicking the list. To remove a rule, mark it as selected by clicking it, then choose *Remove Rule* from the toolbar or the context menu. Editing an existing rule is as simple as double-clicking it, alternatively select *Edit Rule* from the toolbar or the context menu.

All changes you make to the policy page require confirmation before taking effect. This is to make sure you do not at any time have to give more permissions than you would want to, particularly when you are making several new policy decisions at once that depend on each other. To apply the changes, press the *Apply Policy* button on the toolbar. Through the preferences it is also possible to enable an option that causes all changes to immediately take effect.



The inbound traffic policy groups

Inbound policy controls incoming traffic from the Internet and the local network to the firewall. The default inbound policy is one of complete coverage, meaning nothing is let in unless explicitly allowed. The rules you add to the inbound policy groups therefore create exceptions to this policy, effectively creating holes in the firewall through which legitimate traffic may move.

The three inbound policy groups are, from top to bottom, *Allow connections from host*, *Allow service* and *Forward service*.

Allow connections from host

When creating a new rule in the *Allow connections from host* group, the only parameter can you specify is the IP or hostname of the source host. As the name says, adding a host to this group marks the host as a trusted source, all future traffic from the machine will be allowed through the firewall.

Allow service

The *Allow service* group allows for much finer grained control of access. Rules in this group take two parameters, the service and the target. The service can be chosen two ways, either through the drop down list of predetermined services or by explicitly entering the network port number the service uses. In the later case, Firestarter will try to determine the service name itself, but the user is also free to enter the name manually.

The target can be one of three choices; Anyone, LAN clients, or a user specified IP, host or network. Anyone means exactly that, anyone and everyone will be able to access the service in question. LAN clients means that only the clients connected to your internal network are allowed to use the service. The user specified target, can either be an IP address in dotted decimal format, a valid human readable hostname, or a whole network. A network is either specified as *network/netmask* in dotted decimal format, or by using CIDR Notation. In the case where a single IP is specified, the service is effectively *stealthed*, only the target host can see the service.

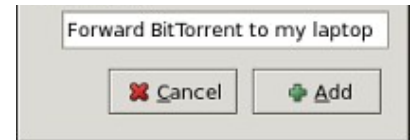
Forward service

The last inbound policy group is *Forward service*. This group is only active if you have enabled Internet connection sharing. When sharing is enabled, a group of computers is seen as a single entity from the outside network's point of view. As the computers all share a single public IP address, in order to provide public services on the LAN machines the firewall has to act as a forwarding relay between the public and private networks.

Like the previous policy group the rules consist of two parts, the service and the target. The service can be chosen in two ways, either by using the drop down list of predetermined services or by directly entering the network port number. This number is the port the firewall will be listening to, when it gets a request on the port it then forwards it to the target. The target is specified as an IP address of the internal network. This is the machine that will be providing the actual service. It possible to specify a different port for this machine that the one used by the firewall, although in most cases they will be same. One use for separate ports on the firewall and the server is that the port do



not have to be symmetric. This means the firewall can listen to a whole range of ports and then forward it all to one single port on the server.



The outbound traffic policy groups

Outbound policy controls outgoing traffic to the Internet from the firewall and any LAN clients. The default outbound policy is permissive. This means you and any clients connected to the local network are able to browse the net, read email, etc. unrestricted. It is also possible to change the policy to a restrictive mode matching that of the default inbound policy. You can toggle between the two modes using the buttons at the top of the outbound policy view.

Permissive mode

The mode marked *Permissive by default, blacklist traffic* is how Firestarter starts out. As explained above, the outgoing traffic is not restricted in this mode and your network applications will work as normal. The policy groups on the outbound policy view serve to black list traffic in this mode, meaning they impose additional restrictions on the otherwise lenient policy.

The first policy group, *Deny connections to host*, is effectively a black list. Rules in this group take a single parameter consisting of an IP address or a valid hostname. The hosts listed in this group are then marked as off limit. One possible use of this policy group is to maintain a list of banned web sites, no one on the local network or using the firewall as a desktop will be able to browse the listed sites.

The *Deny connections from LAN host* group works as a black list for local network clients. Any host listed here will not be able to reach the Internet. This can for example be useful if you want to lock out an internally accessed server but still do not want to impose a restrictive policy for any other clients.

Finally, the *Deny service* group allows the most fine grained control of outbound access. Rules in this group consists of the service and a target. The service is selected as before, the target can be one of four choices; Anyone, Firewall host, LAN clients and a user entered IP, host or network. Adding rules to this group blocks the target from accessing the service in question. The target is selected in the same way as in the *Allow service* group of the inbound policy.

Restrictive mode

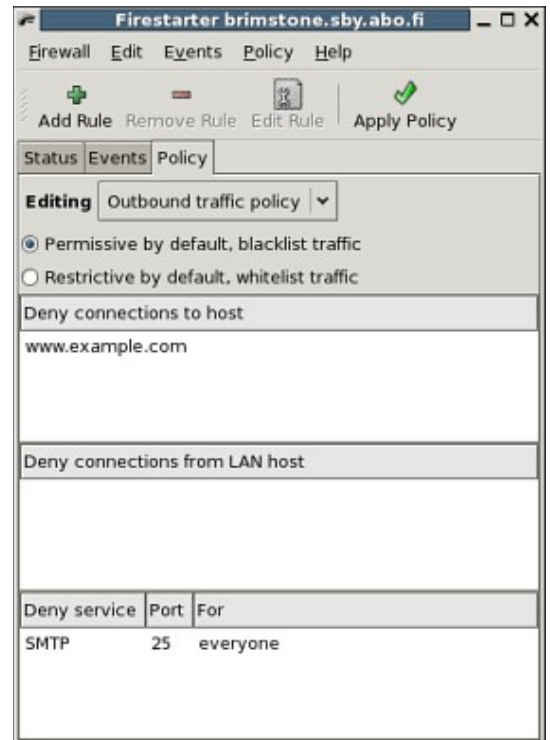
The second outbound policy mode, marked *Restrictive by default, whitelist traffic* is equivalent to the default inbound policy. Nothing is allowed out unless you explicitly create a rule for it in one of the groups. This mode offers maximum protection, but it's also very intrusive, none of your networks applications will work unless you create rules for them.

When you first switch Firestarter to this mode you will see that some rules are already present in the *Allow service* group. These rules allow basic name resolution on the Internet and web browsing. The rules are included so that you do not accidentally create a situation where you are unable to reach help online. Once you are sure you want to use the restrictive mode you can remove them.

Allow connections to host is a whitelist for destinations anyone should be able to reach. Using this group it is for example possible to lock down the machine so that it is only ever able to reach a single web site, something that is desirable when the machine is acting as an kiosk or in other dedicated services.

Allow connections from LAN host gives one single machine on your local network unrestricted access to the Internet.

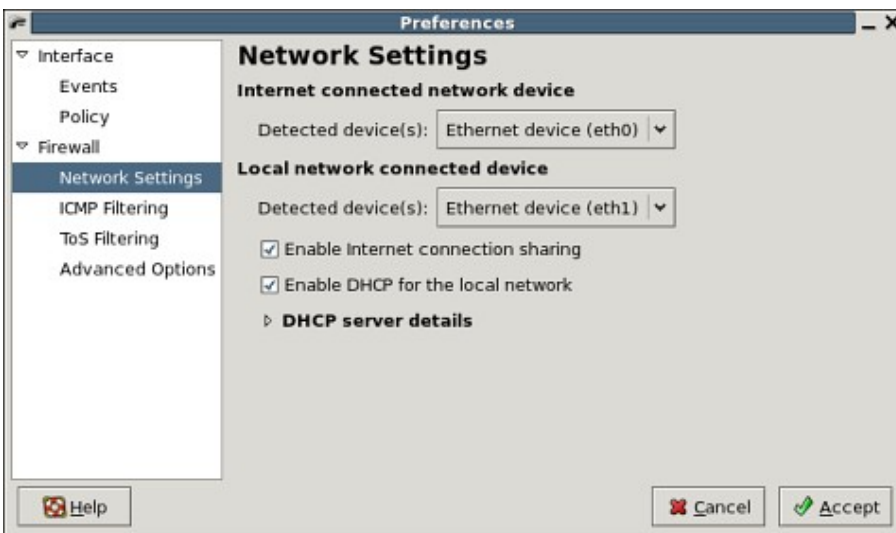
The *Allow service* groups provides fine grain control over outbound access. It is the logical opposite of the *Deny service* group in permissive mode, and the parameters are also the same. Creating rules in this group allows you for example to give exemption to a single machine to use some otherwise forbidden service.



9. The preferences

The Firestarter preferences dialog holds many options that control the behavior of the graphical interface and the firewall. To access the preferences, either click the button on the toolbar of the status page or use the *Edit->Preferences* menu entry.

The preferences are divided into two categories; options that change the interface and options that affect the firewall.



Interface options

The interface options are further divided into three sections, general interface options and options for the events and policy pages.

General interface options

Enable tray icon. When this option is enabled, Firestarter will display a notification icon in the system tray. The icon shows the same state information as on the status page. It will also flash when a new event occurs. Clicking the icon causes the Firestarter main window to hide or reveal itself.

Minimize to tray on window close. This option is meant to be used together with the tray icon option. When enabled, instead of exiting the application when you close the window, the application is hidden from view but actually continues to run in the background. This is similar to how many instant messengers work. The idea is that you do not want the Firestarter window on your desktop all the time, but you still want get alerts when something happens.

Events page options

These options control aspects of the events page.

Skip redundant entries. With this option enabled, the list of blocked connections on the events page will filter out identical consecutive events.

Skip entries where the destination is not the firewall. When enabled, this option causes the program to perform a comparison between the firewall's IP and the destination of the blocked connection. If the two do not match, the entry is not shown in the list. This is useful in a few situations, such as when you have disabled the filtering of broadcast events, but still do not want to see them reported.

This page also contains two lists of hosts and ports that are filtered out from the blocked connections list. When you right-click on an entry on the events page and choose *Disable Event*, the port or source of the entry is added here. Ports and hosts listed here are kept out of the actual firewall event log files, unlike the two previous options which merely hide some of them from view.

Firewall options

The firewall options control some of the more advanced firewalling functions, as well as the choices you made in the initial wizard.

General firewall options

Start/restart firewall on program startup. With this option checked, Firestarter forces a reload of the firewall when you start the graphical interface.

Start/restart firewall on dial-out. This option adds the firewall service to the list of programs to run when a dial-up connection is established. It ensures the firewall is properly restarted when you are assigned a new IP address.

Start/restart firewall on DHCP lease renewal. This option causes the firewall to be restarted when your service provider issues you a new IP address.

Network settings

The options related to the network settings allow you choose the network devices associated with your Internet connection and internal network, if you have one. All network devices in the computer are automatically detected, you only have to choose one from the drop down list of available choices.

Enable Internet connection sharing. This option allows you share the firewall's Internet connection with the other machines on the local network. For more information, please see the chapter on Internet connection sharing.

Enable DHCP for the local network. DHCP is a network service that allows for automatic distribution of the network settings to computers on your local network. For in detail information about configuring the DHCP service, see configuring the DHCP server.

ICMP filtering options

ICMP packets make up a special class of traffic used by many common network utilities, for example *ping* and *traceroute*.

By default Firestarter allows ICMP traffic, although it throttles it somewhat to prevent excessive flooding or Denial of Service attacks. By enabling ICMP filtering you can block these services altogether. Note that blocking a certain ICMP type also prevents you from using it yourself.

Firestarter allows control of the following ICMP types:

- **Echo request and Echo reply**
Used by the ping network tool, but also widely by network enabled games and programs in general. Ping can also be used as a component in various network attacks and information gathering roles.
- **Timestamping**
Timestamping is used by Internet gateways to assess how long a packet destined for a host should remain active before being discarded. Users of the @Home Cable network should not filter timestamping requests, as these are commonly used to maintain statistics tracking and keepalives throughout the networks.
- **Traceroute and MS Traceroute**
These ICMP types are used for pathfinding between networks. Useful for configuring networks for Quality of Service as well as detecting network dropouts. Microsoft Traceroute can safely be filtered if you (or your uplink) do not operate Microsoft DNS servers.
- **Unreachable**
Sometimes used in Fingerprinting attacks, unreachable messages are sent when a destination host is not responding to a request from your machine. Unreachable requests should not be filtered unless you are running DMZ services on your machine.
- **Address Masking**
No longer used since Linux 2.2, you can filter these requests unless you have really old machines on your internal network.
- **Redirection**
Used in inter-network traffic, redirection is useful if you have networks connected in different locations (possibly different countries) connected via a VPN or other networking service. In these cases, the gateways can provide redirection for incoming packets, informing them that the destination host is actually on the same logical network, and should therefore be forwarded. If you are not in this situation, you may filter these packets.
- **Source Quenches**
No longer used since Linux 2.2, you can filter these requests unless you have proprietary UNIX boxes on your internal network that are able to respond to these.

ToS filtering options

Type of Service filtering allows the firewall in some cases to increase the throughput or reliability for certain applications. It does this by re-prioritizing the traffic. Type of Service prioritization needs to be supported by the network you connect to, in practice this limits the area of effect to local networks.

Firestarter can prioritize the traffic for typical workstation and server tasks. Additionally, applications running remotely over the X Window system can be also prioritized. Either the total throughput, connection reliability or the application interactivity can be maximized for the selected work tasks.

Advanced options

The advanced options are mainly for the experienced user.

Preferred packet rejection method. Firestarter can either reject or drop connections that are not allowed by the security policy. When the firewall rejects a connection, it sends an error packet to the source telling it the connection was denied. Dropping a connecting on the other hand does nothing. In this case the source of the denied connection is non the wiser, in some cases it is even impossible to tell whether there is a machine at the firewall's IP at all. Since rejecting connections allows the remote party to map the network services as well as waste your bandwidth, we recommend you keep the default behavior of dropping connections silently.

Block broadcast traffic. This option blocks all network traffic with a destination or source address that marks it as either a

global or local broadcast.

10. Working with policy

A firewall policy is a set of rules that together unambiguously for every connection determine whether it is allowed to pass through the firewall or not. The Firestarter policy is made up of two separate layers, a default policy and a user specified policy.

The default policy

Firestarter tries to provide a safe and user-friendly policy by default. While it protects both the firewall host itself as well as any client hosts connected to a local network from intrusion attempts, it does not impose restrictions on the services that the protected hosts themselves can access. The default policy provides a solid base on which you can choose to implement additional rules, specifying what constitutes both authorized and unauthorized network access.

The default Firestarter policy is as follows:

- New inbound connections from the Internet to the firewall or client hosts are blocked.
- The firewall host is freely allowed to establish new connections.
- All client hosts are allowed to establish new connections to the Internet, but not to the firewall host.
- Traffic from the Internet in response to connection requests from the firewall or client hosts is allowed back in through the firewall.

This policy allows normal Internet usage such as web browsing and e-mail on the secured hosts, but blocks any attempts to access network services from the outside and shields the local network.

Creating new policy

While the default policy can by itself make up the entire firewall rule set, it is often desirable to add your own policies to the set. User specified rules can either relax the default policy or impose further restrictions.

User created policy in Firestarter is grouped according to the class of network traffic affected:

Policy group	Traffic affected
Inbound	Connections originating from the Internet or the local network with the firewall host as the destination.
Outbound	Connections from the firewall host and the local network to the Internet

The reason inbound policy does not affect connections from the Internet to the local network is because such traffic is not directly possible. Only by using the firewall host as a middleman and performing traffic forwarding are local network hosts reachable from the Internet.

Inbound policy

All inbound network traffic that is not in response to a connection established by a secured host is always denied. User created inbound policy is therefore permissive by nature and consist of criteria that when met lift the restrictions on the creation of new incoming connections. Changes to inbound policy are made on the inbound policy section of the policy page in Firestarter.

Outbound policy

The purpose of outbound traffic policy is to specify the types of network traffic that are allowed out from the secured network to the Internet. Firestarter has two modes of operation when it comes to implementing outbound policy, a permissive (which is the default) and a restrictive mode.

The permissive outbound mode, marked "Allow outbound traffic not denied" on the policy page, allow you to specify rules that limit outbound connections.

The restrictive outbound mode on the other hand, marked "Deny outbound traffic not allowed" on the policy page, means you explicitly specify which connections are allowed out. When this mode is enabled for the first time some basic rules are already present in the system. These rules permit the secured hosts to access the DNS, DHCP and HTTP services so that you do not accidentally end up in a situation where you are unable to reach the web or further assistance. Once you know for sure you wish to enable the restrictive outbound policy, you can freely remove these rules.

11. Internet connection sharing

Firestarter has the ability to share the firewall host's Internet connection among all the computers on your local network. This is done through a technique called Network Address Translation, or NAT. To the outside world the cluster of machines will look like a single machine with a single IP address.

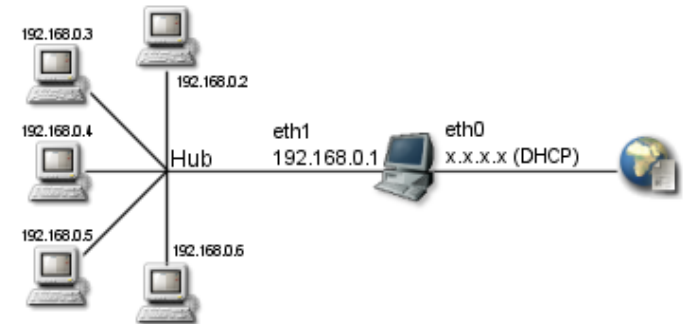
For connection sharing to work you need to have two or more network devices in your firewall. If the local network is set up correctly, enabling connection sharing is as easy as enabling the option in either the firewall wizard or the Firestarter preferences.

The physical setup and network device settings

The procedure for setting up a network using connection sharing is essentially the same whether you have only two computers or a more complex network with hubs or switches connecting multiple computers. For this example we will be assuming that the Internet connected device on the firewall is an Ethernet card, but a modem or ISDN will work too.

The Firewall/gateway machine connected to the Internet will need two network cards and the clients need one each.

The first network card in the firewall, the external interface, will be the one physically connected to the Internet. This card is usually automatically configured with DHCP. The second network card in the firewall, the internal interface, will be connected to the client machines via either a *crossover cable* if the connection goes directly to another computer, or regular cable if you have a hub or switch.



Sharing a connection with a local network

The internal interface of the firewall needs to be statically configured. There are many ways to configure a network interface depending on the distribution you use. Fedora and Red Hat Linux ship with a simple command line tool called *netconfig* and a more sophisticated graphical tool called *system-config-network*. *system-config-network* works better with multiple network cards in the same machine, so we recommend you try it. Other distributions include their own configuration tools, for example in SuSE you would use the Yast program.



Sharing a connection with a single computer

No matter how you decide to configure the network cards, these are settings you should enter:

For the external device (usually eth0):

- Enable dynamic IP configuration (DHCP)
- That's it. You're done, don't touch this card further.

The internal device (usually eth1):

- Disable dynamic IP configuration
- IP address: 192.168.0.1
- Netmask: 255.255.255.0
- Default gateway (IP): <leave empty>

Any changes you make will take effect after a reboot, or more elegantly after a restart of the network services (run `"/etc/init.d/network restart"` as root in most distributions).

Configuring the clients

There are two ways to configure the clients. The more elegant and in the long run easier way is to run a DHCP service on the firewall. A DHCP server distributes the network settings such the IP address, the default gateway, nameservers, etc. at run time to the each client. The alternative to using a DHCP server is to configure every client manually.

Using the DHCP service is as easy as simply enabling it in Firestarter. For more information about the service and how to configure it, refer to the section on configuring the DHCP server.

When using DHCP, the clients need only be configured to use dynamic IP configuration. No other settings need to be changed.

Configuring the clients manually

If you do not wish to use the DHCP service, configure the network devices of the clients to use the following settings:

- Disable dynamic IP configuration
- IP address: 192.168.0.2 to 192.168.0.254, with each client using an unique IP
- Netmask: 255.255.255.0
- Default gateway (IP): 192.168.0.1
- Primary nameserver: Set this to the same nameserver as used on the firewall. You can see the correct setting in the */etc/resolv.conf* file on the firewall.

Restart the network service and you're done.

Testing the Setup

The computers should now be connected and the hardware level configuration complete. To test that everything is ok, try pinging the gateway from the client and vice versa.

Enter the following at the firewall machine console, to test that the gateway can reach the client:

```
[bash]$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) from 192.168.0.1 : 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=255 time=1.37 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=255 time=0.635 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=255 time=0.638 ms

--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2010ms
rtt min/avg/max/mdev = 0.635/0.882/1.375/0.349 ms
[bash]$
```

In case of DHCP, the IP's might be randomly assigned

If it is not working you know that the problem lies with the hardware or network configuration. It is common to get the default gateway setting wrong, so double check it.

At this point:

- The firewall machine should be able to reach the Internet
- The clients and firewall should be able to ping each other
- The clients should be able to reach the Internet if the Internet connection sharing option is enabled in Firestarter.

12. Configuring the DHCP server

DHCP is a network protocol that among other things can dynamically distribute IP addresses and other network setting to computers. When DHCP is used together with Internet connection sharing in Firestarter, the need to individually configure the client machine's TCP/IP settings on the local area network is removed. With DHCP, the process of connecting a new client to the network is as easy as plugging it in.

Specifically, when the DHCP service is enabled in Firestarter, a DHCP server is started on the firewall machine that distributes *DHCP leases* to the client machines. A lease supplies the clients with all the information needed to connect to the network. This information includes a unique IP address for each client, the default gateway, the subnet mask and the domain name servers in use. DHCP leases are always given with the understanding between client and server that the lease is valid only for a limited time.

Enabling the DHCP Service

Note: Firestarter does not itself include a DHCP server, it depends on the underlying system to provide this feature. **The system does not need to have the DHCP server configured, or running.** It is sufficient that the *dhcpcd* program is located on the system, after that Firestarter will manage the DHCP server completely on the user's behalf.

If a DHCP binary is not detected on the system, the DHCP controls will remain inactive

Packages that need to be installed for the Firestarter DHCP service to function

Distribution	Package name	Installed with
Red Hat 9, Fedora Core	dhcp	"yum install dhcp"
Debian	dhcp	"apt-get install dhcp"
Mandrake	dhcp-server	""urpmi dhcp-server"
Gentoo	dhcp	"emerge dhcp"
SuSE	dhcp-server	Manually during system install or from RPM

The DHCP service can be configured from the firewall wizard, or from the preferences. By checking the *"Enable DHCP for local network"* option on the connection sharing setup screen the service is activated upon the completion of the wizard or saving of the preferences. Note that DHCP is only available in conjunction with Internet connection sharing, and the DHCP service is only provided to machines connected to the local area network. This way Firestarter does not interfere with the DHCP systems of Internet service providers.

Optionally a few details related to the DHCP server can be configured from the graphical interface. By clicking the *"DHCP server details"* label or arrow, new options become available. It is possible to change the range of IP addresses that is distributed to clients by changing the values of the *lowest* and *highest IP address to assign* controls. By default Firestarter will distribute IP addresses in the 192.168.0.100 to 192.168.0.254 range. The IP range can be freely manipulated, but must fall within the constraints created by the network address and netmask of the local area network connected network adapter.

Finally, the *name server* can be specified freely. Both dotted decimal IP addresses and hostnames of name servers are accepted. It is also possible to specify several servers at once by entering a comma separated list of servers. A special value of *<dynamic>* can also be specified, in which case the name servers are determined dynamically at run time from the firewall server's network settings. This is especially useful if the firewall server itself is using DHCP and the provider's name servers are subject to change. The default is to determine the name servers dynamically.

Creating Your Own DHCP Configurations

The option to *keep an existing DHCP configuration* is available if Firestarter detects that a DHCP configuration is already present on the system. This option has several uses. If you have manually configured a DHCP server on your system previously, it is possible to have Firestarter only manage the server process without touching the actual configuration. Alternatively, you can take the Firestarter generated DHCP configuration and extend it, safe in the knowledge that Firestarter will not overwrite it later on without your permission. There exists a number of interesting DHCP related features you can configure yourself, including whitelisting based on MAC hardware addresses, specifying static IPs for specific hosts, dividing machines into different classes and configurations, providing boot images for network terminals etc. See the *dhcpcd.conf* and *dhcpcd man* pages for more information.

The Firestarter generated DHCP configuration is stored as */etc/dhcpcd.conf*.

13. Advanced topics

This chapter gathers topics of interest mainly to advanced users. It includes information on how Firestarter works under the hood.

- Persistence of the Firewall
What happens when you exit the application, or if it crashes, or when you reboot the system? This chapter tries to sort out these questions. Also included is information on how to run the Firestarter firewall as a system service.
- Kernel Requirements
When you compile your own kernel it is important to include the necessary modules Firestarter requires.

14. Persistence of the firewall

A frequently asked question is, what is the state of the firewall when you are not running the Firestarter program? The answer is that it depends. If you installed the Firestarter from a binary package such RPM or Deb, the firewall will be running all the time (dial-up users excluded) and independent of the graphical interface, even after a reboot. In these cases the firewall is registered as a system service and can be manipulated using the standard Linux system service and runlevel management tools.

If you compiled and installed Firestarter from source and are using a Fedora, Red Hat, SuSE or Mandrake based Linux distribution, you have the option of installing a system init script. See the installation instructions on how to do so. Other distributions include their own customized init scripts in the Firestarter binary packages they provide. Once the init script is installed and activated, Firestarter is running as a system service.

If you compiled from source and are not using an init script the firewall is active from the moment you run Firestarter to the next reboot. See the notes about starting the firewall manually below for increasing your firewall coverage.

In addition to the behaviors described above the usage of a DHCP daemon further extends the coverage. When the network device bound to the DHCP service is assigned an IP address (either when connecting for the first time or on a lease renewal) the firewall is either started or refreshed. Note that this occurs even if the firewall was stopped, either from the Firestarter program or from the init scripts. Currently this service is provided when using either the DHCPD or dhclient programs (this covers pretty much any modern distribution). When using DHCP it is therefore not strictly necessary to have an init script.

The System Init Scripts

Firestarter comes with a SysV style init script for managing the firewall. The script provides the following functions:

- start: starts the firewall.
- stop: stops the firewall.
- restart: restarts the firewall.
- locks: locks the firewall.
- status: reports the state of the firewall.

See the manual entry for the status page for information about the various firewall states.

The functions can be invoked by appending them as parameters to the script. For example, on a Red Hat / Mandrake distribution you can start the firewall by running `/etc/init.d/firestarter start`. Most distributions also include tools, like `chkconfig`, to manage the service scripts. These tools allow you to change the boot priority and many other parameters of the services.

Managing the Firewall Manually From the Console

The Firestarter program accepts a number of command line parameters for manipulating the firewall. Run `firestarter --help` for the complete list of options. If you installed from RPM make sure you're running `/usr/sbin/firestarter` and not `/usr/bin/firestarter` as the later is merely a wrapper.

Dial-up Issues

The firewall must be started after you have established your connection to your ISP. In the Firestarter wizard there is an option to automatically start the firewall on dial-out. This option does not work with some dialers. For example, if you are using the `kppp` dialer application you will have to set up the dialer to start the firewall after a connection is established. Kppp includes an option to launch scripts when a connection is established which does the job nicely.

15. Kernel requirements

Firestarter requires certain Linux kernel subsystems to be present in order to function properly. This is not an issue with vendor provided kernels, as all Linux distributions provide the required functionality out of the box these days. However, if you are creating your own customized kernels the following section is likely of interest to you.

Configuring the kernel

These instructions are meant for the Linux 2.6 kernel. Note that Firestarter 0.9.x also works with the older 2.4 kernel, in which case the basic configuration is the same but the kernel options are laid out differently in the configurator. It is assumed the user already knows how to compile a kernel, see the Kernel Build HOWTO if you require help in this regard.

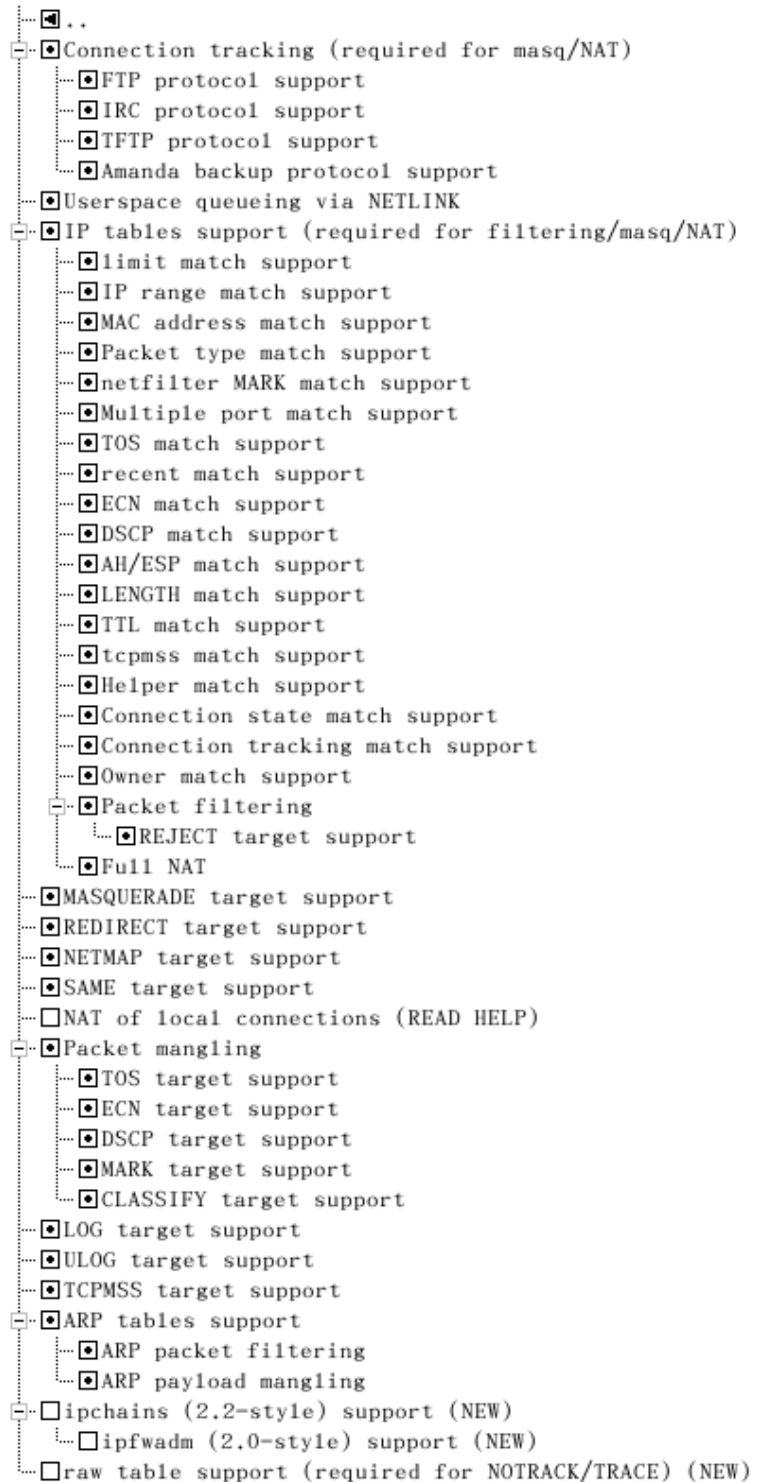
All of the kernel options affecting Firestarter are located under the *Device Drivers->Networking support* menu. First, the option *Networking support*, must be enabled. Then, under the submenu *Networking options->Network packet filtering*, the *Network packet filtering* option must be enabled. Finally, the submenu *IP: Netfilter configuration* presents you with a long list of Netfilter modules, the exact contents of which depends on your specific kernel revision. Here you have the option of enabling the features as either modules or compiled statically into the kernel. It is recommended that you compile all of the features as modules unless you are specifically creating a kernel without module support. This way, when you are not using a particular feature it will not use any memory.

Firestarter versions prior to 1.0 output error messages to the console if you compile the features statically into the kernel and not as modules. However, if all the required features are present in the kernel, Firestarter will work fine anyway.

All of the Netfilter feature are however not required, although we recommend you enable them as modules in any case as this allows future versions of Firestarter to make use of them and there is no harm in doing so. At the very least, the *Connection tracking*, *IP tables*, *Connection state match support*, *Connection tracking match support*, *Packet filtering*, *Full NAT* and the *LOG target support* features must be present. Some of the other features, when detected at run time in the kernel, enable additional functionality in Firestarter. The only features we recommend you do not include are the legacy *ipchains support* and *ipfwadm support* options.

Kernel features location reference

- Device drivers
 - Networking support
 - [y] Networking support
 - Networking options
 - Network packet filtering
 - [y] Network packet filtering
 - IP: Netfilter Configuration (*)
 - [y/m] Connection tracking
 - [y/m] IP tables support
 - [y/m] Connection state match support
 - [y/m] Connection tracking match support
 - [y/m] Packet filtering
 - [y/m] Full NAT
 - [y/m] LOG target support



Picture: Recommended configuration for Linux kernel version 2.6

(*) We recommend you enable everything except *ipchains support* and *ipfwadm support* as modules under this menu

16. Virtual Private Networking

Firestarter 1.0 does not support VPN configurations without some tweaking. VPN capability in Firestarter is currently planned for version 1.1.

How to use the VPN workarounds in Firestarter 1.0

Copy the lines specific to your VPN solution listed below, and paste them into the `/etc/firestarter/user-pre` file on the firewall host. Restarting the firewall, for example by executing `"/etc/firestarter/firewall.sh start"`, commits the new settings.

Microsoft VPN clients

The widely used VPN solution for Microsoft Windows machines is based on the PPTP protocol. The following lines allow PPTP clients on the Firestarter administered local network to connect to remote servers:

```
# Forward PPTP VPN client traffic
$IPT -A FORWARD -i $IF -o $INIF -p tcp --dport 1723 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $IF -o $INIF -p 47 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $INIF -o $IF -p 47 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

Running a Microsoft VPN server

If you want to run a PPTP VPN server on your internal network and allow remote clients to connect to it, the firewall must be told to forward the requests to the server:

```
# Forward PPTP VPN connections to internal server
SERVER=192.168.0.100 # Internal VPN server

$IPT -A FORWARD -i $IF -o $INIF -p tcp --dport 1723 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -t nat -A PREROUTING -i $IF -p tcp --dport 1723 -j DNAT --to $SERVER
$IPT -A FORWARD -i $IF -o $INIF -p 47 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -t nat -A PREROUTING -i $IF -p 47 -j DNAT --to $SERVER
```

Note that you must change the `SERVER` variable from `192.168.0.100` to the actual internal IP of your VPN server.

OpenVPN

OpenVPN is an easy to use cross-platform VPN solution that is also Open Source. If OpenVPN is to be used on the computer that Firestarter is running on, traffic must be allowed to and from the OpenVPN virtual interface with the following lines:

```
# Allow traffic on the OpenVPN interface
$IPT -A INPUT -i tun+ -j ACCEPT
$IPT -A OUTPUT -o tun+ -j ACCEPT
```

OpenVPN requires no configuration changes if it is used on the local network.

Cisco VPN and Nortel Contivity clients

Cisco, Nortel and other IPsec based VPN solutions require the following workaround:

```
# Forward Cisco VPN client traffic
$IPT -A FORWARD -i $IF -o $INIF -p udp --dport 500 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $IF -o $INIF -p tcp --dport 500 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $IF -o $INIF -p 50 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -i $INIF -o $IF -p 50 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

Firestarter 1.0 works out of the box with the Cisco VPN clients when *Transparent Tunnelling* is enabled client side. In this mode the client embeds the IPsec traffic into either TCP or UDP packets, which Firestarter is able to deal with as any other form of traffic.

However, tunneling is not always possible, depending on both the client version and the remote server's capabilities. In that case the above workaround must be used.

17. Frequently asked questions

This chapter contains answers to commonly asked questions, as well as various tips and tricks for Firestarter.

Questions:

- How can I get Firestarter to load automatically when I log in as a regular user?
- How do you specify a range of IPs or use wildcards in the rules?
- Do I have to start Firestarter after I have rebooted?
- How can I test if the firewall is working for sure?

Is your question not answered here? Contact us!

Q: How can I get Firestarter to load automatically when I log in as a regular user?

Normally when you start Firestarter by clicking an icon or manually from a terminal, the system will prompt you for your root user's password. However, this is a bit of a hassle, especially if you want to run Firestarter all the time when logged in. In that case Firestarter can be loaded in the background when you log in with your regular user, without asking a password and minimized to the system tray (pictured right).



Giving the user permission to launch Firestarter without the root password

In order for a regular user to be able to launch Firestarter, the user must be given additional privileges. Edit your `/etc/sudoers` file in your favorite text editor and add the following line at the end:

```
username ALL= NOPASSWD: /usr/bin/firestarter
```

Simply replace `username` with whatever your login is. The specified user is now able to launch Firestarter without being prompted for a password using the command `sudo firestarter`.

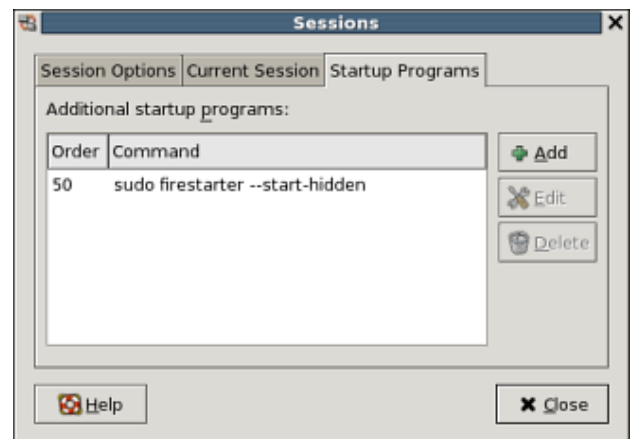
A note on the security aspects: This method makes a trade off in local security for convenience. If your user account becomes compromised the attacker will be able to control the firewall. However this method is preferable to having a shared root user password in a multiuser setting. It is also preferable if the alternative is to not run Firestarter at all.

Launching Firestarter minimized to the tray on login

Open up your GNOME menu, select *Preferences* followed by *Sessions*. Switch to the *Startup programs* tab, pictured right.

Click *Add* and enter `sudo firestarter --start-hidden` as the startup command. Click *OK* and you're done.

Now, whenever you log in as your regular user, Firestarter will load into the system tray automatically. Simply click the tray icon to bring up the main Firestarter interface.



Q: How do you specify a range of IPs or use wildcards in the rules?

Wherever in Firestarter a single numerical IP address can be inputted as part of a policy rule, a human readable hostname or a *network* identifier can also be used. This last form allows you to apply rules to a range of IP addresses.

A range is entered as either *address/netmask*, for example `192.168.0.1/255.255.255.0`, or more commonly in CIDR format as `192.168.0.1/24`.

The CIDR address consists of a standard dotted format 32-bit IP address and a postfix of the number of network identifying bits. This might sound confusing, and it is, but it is the only valid way to group IP addresses on the Internet.

Luckily you don't have to break out your pocket calculator to work out an IP range, as there exists many IP calculators online.

CIDR range notation examples

CIDR format	First host	Last host	Number of hosts
192.168.0.1/24	192.168.0.1	192.168.0.254	254
192.168.0.1/25	192.168.0.1	192.168.0.126	126
192.168.0.1/26	192.168.0.1	192.168.0.62	62
192.168.0.1/27	192.168.0.1	192.168.0.30	30
192.168.0.1/29	192.168.0.1	192.168.0.9	6
192.168.0.9/29	192.168.0.9	192.168.0.14	6
192.168.0.10/30	192.168.0.10	192.168.0.11	2
10.0.0.0/8	10.0.0.1	10.255.255.254	16777214
10.0.1.17/28	10.0.1.17	10.0.1.30	14

Q: Do I have to start Firestarter after I have rebooted?

Usually, no. When Firestarter is installed from a package, the firewall is running as a service. You can query the status of the service by executing `/etc/init.d/firestarter status`. The exemption to this is Gentoo users, dial-up users in some cases and persons who have installed from source and not registered the Firestarter system service.

For an in-depth answer, see the section on persistence of the firewall.

Q: How can I test if the firewall is working for sure?

The only way to know for sure if your firewall coverage is complete is for an outside party to test it. **You can not run nmap or some other network tool to test the firewall from the firewall host itself.**

There are many free sites on the Internet that will provide a remote scan of your system. Here are a few that we have found useful, as well as the expected result with the Firestarter default policy loaded:

- Sygate Security Scan
The scan will report *Unable to detect any running services!*
- Shields Up!
The scan will report all ports as *stealtd*

Why you might not be getting the results you expect

If some specific port is reported as *Closed* instead of *Stealthed* by Shields Up, your Internet service provider is probably blocking the port before the scanner even connects to your machine. This is typical for ports such as 25 (SMTP) and 80 (HTTP) that your ISP prohibits you from running services on.

If you have a DSL or cable modem box that provides *Network Address Translation* services, it is possible that the scan does not reflect the status of Firestarter but that of the box.

18. Developing Firestarter

Firestarter is an Open Source project and therefore open to contributions from the community. If you want to help develop Firestarter you should start by joining the Firestarter mailing list and discussing your proposal with us. You can also contact us main developers directly.

You do not have to be a programmer to help out. We are looking for all sorts of help:

- Documentation
We need people to complete the documentation, write tutorials, explore advanced subjects, etc.
- Art
We are always looking for someone to create new icons, logos and pictures for the website. Please contact us directly if you have the skill and want to help.
- Translations
Firestarter is translated into around 40 languages, but the translations need to constantly be maintained which requires a lot of work. Translations are coordinated with the help of the GNOME Translation Project, if you want to help you should start by join your language's translation team.
- Creating distribution packages
Almost all of the Firestarter binary packages are contributed by users. If your favorite distribution is not represented you can help out by creating the package yourself and submitting it to us.
- <Your skill here>
You probably know best exactly how you can help and apply your unique skills.

Persons who wish to work as programmers on the project should track the CVS development version. CVS access is provided through the GNOME CVS server and open to all GNOME developers.

So join in, have fun, gain valuable experience and a bit of online prestige.

19. Getting Firestarter from CVS

Firestarter is developed with the aid of a version control system called CVS. It allows multiple developers to work on the code simultaneously. By checking out the code from the CVS repository you are getting the bleeding edge development version of Firestarter. CVS usage is intended for testers, developers and users that need the most recent code available.

CVS releases of Firestarter are not stable, release quality code. If you are interested in obtaining the latest, stable release - you can download a tarball of the source code from the Firestarter homepage. It is not recommended to run the CVS version if you are not prepared to submit feedback and help us improve it.

The Firestarter code is maintained in the GNOME project's CVS server. In order to check out a working copy, enter the following in a terminal:

```
[bash]$ export CVSROOT=:pserver:anonymous@anoncvs.gnome.org:/cvs/gnome'
[bash]$ cvs login
(Logging in to anonymous@cvs.firestarter.sourceforge.net)
CVS password:
```

Simply press enter here, there is no password.

```
[bash]$ cvs -z3 checkout firestarter
```

The source code will be downloaded to the 'firestarter' subdirectory. To compile, change to the 'firestarter' subdirectory and type the following commands:

```
[bash]$ ./autogen.sh --sysconfdir=/etc
[bash]$ make
```

You might also have to download the gnome-common module from CVS (`cvs -z3 checkout gnome-common`) and install it before autogen will run successfully. If all goes well, su to root and type:

```
[bash]$ make install
```

Bringing a CVS version up to date

If you already have a recent CVS Firestarter checkout from the CVS server, you can simply enter the following commands to update it to the current version:

```
[bash]$ cd [copy_of_current_CVS_source_code]/
[bash]$ export CVSROOT=:pserver:anonymous@anoncvs.gnome.org:/cvs/gnome'
[bash]$ cvs login
(Logging in to anonymous@cvs.firestarter.sourceforge.net)
CVS password:
```

Again, simply press enter here, there is no password.

```
[bash]$ cvs -z3 update -Pd firestarter
```

The CVS repository can also be browsed online.

20. License

Firestarter is Copyrighted © 1998-2004 by Tomas Junnonen and licensed under the GPL.

The Firestarter 1.0 manual is Copyrighted © 2004 by Tomas Junnonen.

The manual for versions up to 0.9.x Copyrighted © 2004 by Tomas Junnonen & Paul Drain.

The images on this site, with the exception of the Firestarter logo, can be re-used under GNU Free Documentation License.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You may obtain a copy of the GNU General Public License from the Free Software Foundation by visiting their Web site or by writing to

Free Software Foundation, Inc.
59 Temple Place - Suite 330
Boston, MA 02111-1307
USA

