Site Search

Mailing List Archive Search

Home

Important Notices

Download
Documentation
Development

Support
Mirrors
Other Links

Introduction

Tom Eastep

2020/03/06

Table of Contents

Introduction

The information in this document applies only to 4.3 and later releases of Shorewall.

Glossary

- Netfilter - the packet filter facility built into the 2.4 and later Linux kernels.
- ipchains - the packet filter facility built into the 2.2 Linux kernels. Also the name of the utility program used to configure and control that facility. Netfilter can be used in ipchains compatibility mode.
- iptables - the utility program used to configure and control Netfilter. The term *"iptables"* is often used to refer to the combination of iptables+Netfilter (with Netfilter not in ipchains compatibility mode).
- iptables-restore - a program included with iptables that allows for atomic installation of a set of Netfilter rules. This is a much more efficient way to install a rule set than running the iptables utility once for each rule in the rule set.
- ifconfig - An obsolete program included in the net-utils package. ifconfig was used to configure network interfaces.
- route - An obsolete program included in the net-utils package. route was used to configure routing.
- ip - A program included in the iproute2 package. ip replaces ifconfig and route in modern Linux systems.
- tc - A program included in the iproute2 package. tc is used to configure QOS/Traffic Shaping on Linux systems.

What is Shorewall?

The Shoreline Firewall, more commonly known as *"Shorewall"*, is high-level tool for configuring Netfilter. You describe your firewall/gateway requirements using entries in a set of configuration files. Shorewall reads those configuration files and with the help of the iptables, iptables-restore, ip and tc utilities, Shorewall configures Netfilter and the Linux networking subsystem to match your requirements. Shorewall can be used on a dedicated firewall system, a multi-function gateway/router/server or on a standalone GNU/Linux system. Shorewall does not use Netfilter's ipchains compatibility mode and can thus take advantage of Netfilter's connection state tracking capabilities.

Shorewall is not a daemon. Once Shorewall has configured the Linux networking subsystem, its job is complete and there is no *"Shorewall process"* left running in your system. The /sbin/shorewall program can be used at any time to monitor the Netfilter firewall.

Shorewall is not the easiest to use of the available iptables configuration tools but I believe that it is the most flexible and powerful. So if you are looking for a simple point-and-click set-and-forget Linux firewall solution that requires a minimum of networking knowledge, I would encourage you to check out the following alternatives:

- UFW (Uncomplicated Firewall)
- ipcop

If you are looking for a Linux firewall solution that can handle complex and fast changing network environments then Shorewall is a logical choice.

Shorewall Concepts

The configuration files for Shorewall are contained in the directory `/etc/shorewall` -- for simple setups, you will only need to deal with a few of them.

Shorewall views the network where it is running as being composed of a set of *zones*. Zones are declared and given a type in the `/etc/shorewall/zones` file.Here is the `/etc/shorewall/zones` file from the three-interface sample:

```
#ZONE   TYPE    OPTIONS                 IN                      OUT
#                                       OPTIONS                 OPTIONS
fw      firewall
net     ipv4
loc     ipv4
dmz     ipv4
```

Documentation

Frequently Used Articles
- FAQs
- Manpages
- Configuration File Basics
- Beginner Documentation
- Troubleshooting

Shorewall 4.4/4.5/4.6 Documentation

Shorewall 4.0/4.2 Documentation

Shorewall 5.0/5.1/5.2 HOWTOs and Other Articles
- 6to4 and 6in4 Tunnels
- Accounting
- Actions
- Aliased (virtual) Interfaces (e.g., eth0:0)
- Anatomy of Shorewall
- Anti-Spoofing Measures
- AUDIT Target support
- Bandwidth Control
- Blacklisting/Whitelisting
- Bridge/Firewall
- Building Shorewall from GIT
- Commands
- Compiled Programs
- Configuration File Basics
- DHCP
- DNAT
- Docker
- Dynamic Zones
- ECN Disabling by host or subnet
- Events
- Extension Scripts
- Fallback/Uninstall
- FAQs
- Features
- Fool's Firewall
- Forwarding Traffic on the Same Interface
- FTP and Shorewall
- Helpers/Helper Modules
- Installation/Upgrade
- IPP2P
- IPSEC
- Ipsets
- IPv6 Support
- ISO 3661 Country Codes
- Kazaa Filtering
- Kernel Configuration
- KVM (Kernel-mode Virtual Machine)
- Limiting Connection Rates
- Linux Containers (LXC)
- Linux-vserver
- Logging
- Macros
- MAC Verification
- Manpages
- Manual Chains
- Masquerading
- Multiple Internet Connections from a Single Firewall
- Multiple Zones Through One Interface
- My Shorewall Configuration
- Netfilter Overview
- Network Mapping
- No firewalling of traffic between bridge port
- One-to-one NAT
- Operating Shorewall
- OpenVPN
- OpenVZ
- Packet Marking
- Packet Processing in a Shorewall-based Firewall
- 'Ping' Management
- Port Forwarding
- Port Information
- Port Knocking (deprecated)
- Port Knocking, Auto Blacklisting and Other Uses of the 'Recent Match'
- PPTP
- Proxy ARP
- QuickStart Guides
- Release Model
- Requirements
- Routing and Shorewall
- Routing on One Interface
- Samba
- Shared Shorewall/Shorewall6 Configuration
- Shorewall Events
- Shorewall Init
- Shorewall Lite

Home

Important Notices

Download
Documentation
Development

Support
Mirrors
Other Links

Note that Shorewall recognizes the firewall system as its own zone. The name of the zone designating the firewall itself (usually 'fw' as shown in the above file) is stored in the shell variable $*FW* which may be used throughout the Shorewall configuration to refer to the firewall zone.

The simplest way to define the hosts in a zone is to associate the zone with a network interface using the /etc/shorewall/interfaces file. In the three-interface sample, the three zones are defined using that file as follows:

```
#ZONE   INTERFACE       OPTIONS
net     NET_IF          tcpflags,dhcp,nosmurfs,routefilter,logmartians,sourceroute=0,physical=et
loc     LOC_IF          tcpflags,nosmurfs,routefilter,logmartians,physical=eth1
dmz     DMZ_IF          tcpflags,nosmurfs,routefilter,logmartians,physical=eth2
```

The above file defines the *net* zone as all IPv4 hosts interfacing to the firewall through eth0, the *loc* zone as all IPv4 hosts interfacing through eth1 and the *dmz* as all IPv4 hosts interfacing through eth2. The interface names shown in the INTERFACE column are *logical* names which are used throughout the configuration to refer to the individual interfaces. The actual interface names are specified using the **physical** option. It is important to note that the composition of a zone is defined in terms of a combination of addresses **and** interfaces. When using the /etc/shorewall/interfaces file to define a zone, all addresses are included; when you want to define a zone that contains a limited subset of the IPv4 address space, you use the /etc/shorewall/hosts file or you may use the nets= option in /etc/shorewall/interfaces:

```
#ZONE   INTERFACE       OPTIONS
net     NET_IF          tcpflags,dhcp,nosmurfs,routefilter,logmartians,sourceroute=0,physical=et
loc     LOC_IF          tcpflags,nosmurfs,routefilter,logmartians,physical=eth1,nets=172.20.1.0/
dmz     DMZ_IF          tcpflags,nosmurfs,routefilter,logmartians,physical=eth2
```

The above file defines the *net* zone as all IPv4 hosts interfacing to the firewall through eth0 *except* for 192.168.0.0/23, the *loc* zone as IPv4 hosts 192.168.0.0/24 interfacing through eth1 and the *dmz* as IPv4 hosts 192.168.1.0/24 interfacing through eth2 (Note that 192.168.0.0/24 together with 192.168.1.0/24 comprises 192.168.0.0/23).

Rules about what traffic to allow and what traffic to deny are expressed in terms of zones.

- You express your default policy for connections from one zone to another zone in the /etc/shorewall/policy file. The basic choices for policy are:

  - ACCEPT - Accept the connection.

  - DROP - Ignore the connection request.

  - REJECT - Return an appropriate error to the connection request.

  Connection request logging may be specified as part of a policy and it is conventional (and highly recommended) to log DROP and REJECT policies.

- You define exceptions to these default policies in the /etc/shorewall/rules file.

- You only need concern yourself with connection requests. You don't need to define rules for handling traffic that is part of an established connection and in most cases you don't have to worry about how related connections are handled (ICMP error packets and related TCP connection requests such as used by FTP).

For each connection request entering the firewall, the request is first checked against the /etc/shorewall/rules file. If no rule in that file matches the connection request then the first policy in /etc/shorewall/policy that matches the request is applied. If there is a default action defined for the policy in /etc/shorewall/shorewall.conf then that action is invoked before the policy is enforced. In the standard Shorewall distribution, the DROP policy has a default action called **Drop** and the REJECT policy has a default action called **Reject**. Default actions are used primarily to discard certain packets silently so that they don't clutter up your log.

The /etc/shorewall/policy file included with the three-interface sample has the following policies:

```
#SOURCE    DEST       POLICY     LOGLEVEL    LIMIT
loc        net        ACCEPT
net        all        DROP       info
all        all        REJECT     info
```

In the three-interface sample, the line below is included but commented out. If you want your firewall system to have full access to servers on the Internet, uncomment that line.

```
#SOURCE    DEST       POLICY     LOGLEVEL    LIMIT
$FW        net        ACCEPT
```

The above policies will:

- Allow all connection requests from your local network to the Internet

- Drop (ignore) all connection requests from the Internet to your firewall or local networks; these ignored connection requests will be logged using the *info* syslog priority (log level).

- Optionally accept all connection requests from the firewall to the Internet (if you uncomment the additional policy)

- reject all other connection requests; these rejected connection requests will be logged using the *info* syslog priority (log level).

A word about Shorewall logging is in order. Shorewall does not have direct control over where its messages are logged; that is determined by the configuration of the logging daemon (syslog, rsyslog, syslog-ng, ulogd, etc.). The LOGFILE setting in /etc/shorewall/shorewall.conf tells Shorewall *where to find the log*; it doesn't determine where messages are logged. See the Shorewall logging article for more information.

To illustrate how rules provide exceptions to policies, suppose that you have the polices listed above but you want to be able to connect to your firewall from the Internet using Secure Shell (SSH). Recall that SSH connects using TCP port 22. You would add the following rule to /etc/shorewall/rules:

```
#ACTION    SOURCE       DEST     PROTO     DPORT
ACCEPT     net          $FW      tcp       22
```

So although you have a policy of ignoring all connection attempts from the net zone (from the Internet), the above exception to that policy allows you to connect to the SSH server running on your firewall.

Because Shorewall makes no assumptions about what traffic you want accepted, there are certain rules (exceptions) that need to be added to almost any configuration.

- Shorewall on a Laptop
- Shorewall Perl
- Shorewall Setup Guide
- SMB
- SNAT
- Split DNS the Easy Way
- Squid with Shorewall
- Starting/stopping the Firewall
- Static (one-to-one) NAT
- Support
- Tips and Hints
- Traffic Shaping/QOS - Simple
- Traffic Shaping/QOS - Complex
- Transparent Proxy
- UPnP
- Upgrade Issues
- Upgrading to Shorewall 4.4 (Upgrading Debian Lenny to Squeeze)
- VPN
- VPN Passthrough
- White List Creation
- Xen - Shorewall in a Bridged Xen DomU
- Xen - Shorewall in Routed Xen Dom0

- The QuickStart guides point to pre-populated files for use in common setups and the Shorewall Setup Guide shows you examples for use with other more complex setups.
- Again, to keep your firewall log from filling up with useless noise, Shorewall provides common actions that silently discard or reject such noise before it can be logged. As with everything in Shorewall, you can alter the behavior of these common actions (or do away with them entirely) as you see fit.

Home

Important Notices

Download
Documentation
Development

Support
Mirrors
Other Links

Compile then Execute

Shorewall uses a "compile" then "execute" approach. The Shorewall configuration compiler reads the configuration files and generates a shell script. Errors in the compilation step cause the script to be discarded and the command to be aborted. If the compilation step doesn't find any errors then the shell script is executed.

The 'compiled' scripts are placed by default in the directory `/var/lib/shorewall` and are named to correspond to the command being executed. For example, the command **/sbin/shorewall start** will generate a script named `/var/lib/shorewall/.start` and, if the compilation is error free, that script will then be executed. If the script executes successfully, it then copies itself to `/var/lib/shorewall/firewall`. When an **/sbin/shorewall stop** or **/sbin/shorewall clear** command is subsequently executed, `/var/lib/shorewall/firewall` is run to perform the requested operation.

The AUTOMAKE option in /etc/shorewall/shorewall.conf may be set to automatically generate a new script when one of the configuration files is changed. When no file has changed since the last compilation, the **/sbin/shorewall start**, **/sbin/shorewall reload** and **/sbin/shorewall restart** commands will simply execute the current `/var/lib/shorewall/firewall` script.

Shorewall Packages

Shorewall 4.5 and later consists of six packages.

1. **Shorewall-core**. All of the other packages depend on this one.
2. **Shorewall**. This package must be installed on at least one system in your network. It contains everything needed to create an IPv4 firewall.
3. **Shorewall6**. This package requires the Shorewall package and adds those components needed to create an IPv6 firewall.
4. **Shorewall-lite**. Shorewall allows for central administration of multiple IPv4 firewalls through use of Shorewall lite. The full Shorewall product is installed on a central administrative system where compiled Shorewall scripts are generated. These scripts are copied to the firewall systems where they run under the control of Shorewall-lite.
5. **Shorewall6-lite**. Shorewall allows for central administration of multiple IPv6 firewalls through use of Shorewall6 lite. The full Shorewall and Shorewall6 products are installed on a central administrative system where compiled Shorewall scripts are generated. These scripts are copied to the firewall systems where they run under the control of Shorewall6-lite.
6. **Shorewall-init**. May be installed with any of the other firewall packages. Allows the firewall to be closed prior to bringing up network interfaces. It can also react to interface up/down events.

License

This program is free software; you can redistribute it and/or modify it under the terms of Version 2 of the GNU General Public License as published by the Free Software Foundation.

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 detail.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Please report errors on this site to the Webmaster

Site design by Connexer Ltd.