



**Trialware Version**

J. River, Inc.



## Acknowledgements

ICE.Block: Firewall protection for networks on the Internet.

© 1995-1997 J. River, Inc. All rights reserved. No part of this manual or the software described in it may be reproduced, translated, transmitted, or stored in a retrieval system, in any form or by any means, without the prior written consent of J. River, Inc.

This manual and the software described in it are provided under license and may be used or copied only in accordance with the terms of the license. The information in this manual is provided for informational purposes only and is subject to change without notice. J. River, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

ICE, ICE.Block, ICE.TCP, ICE.TEN, ICE.Off.Site, and ICE.NFS are trademarks of J. River, Inc.

## Limited Warranty

The software and related manual are provided "as is." J. River, Inc. makes no representations or warranties with respect to the software and manual and disclaims any express or implied warranties of merchantability or fitness for any particular purpose. J. River, Inc. reserves the right to make changes to any and all parts of the software at any time without notice.

J. River, Inc. warrants the disk(s) on which the software is recorded and accompanying written materials to be free from defects in materials and workmanship under normal use for a period of thirty (30) days from the date they were shipped from J. River, Inc. J. River, Inc. makes no other warranty.

J. River, Inc.

125 North First Street • Minneapolis, Minnesota 55401

Telephone: 612-339-2521 • Fax: 612-339-4445

E-mail: [info@jriver.com](mailto:info@jriver.com) • World Wide Web: <http://www.jriver.com>



ICE.BLOCK (TRIALWARE)

# Table of Contents

## **Chapter 1: Introduction ..... 5**

The ICE Family of Products .....	5
Overview of Network Security .....	6
Overview of ICE.Block .....	7
Requirements and Trialware Agreement .....	8
Purchasing a Licensed Version of ICE.Block .....	9
About This Manual .....	10
Technical Support .....	10
Platinum Technical Support .....	11

## **Chapter 2: Installation ..... 13**

Preparing for Installation .....	13
Creating an Installation Disk .....	14
Obtaining a Trialware Activation Key .....	14
Installing ICE.Block on SCO Systems .....	16
Step 1: Enter System Maintenance Mode .....	16
Step 2: Install ICE.Block from the Program Disk .....	16
Step 3: Re-link the Kernel .....	16
Step 4: Enter the Activation Key .....	16
Step 5: Reboot the System .....	17
Removing ICE.Block from SCO Systems .....	17
Updating ICE.Block on SCO Systems .....	17
Installing ICE.Block on UnixWare or Esix Systems .....	18
Step 1: Install ICE.Block from the Program Disk .....	18
Step 2: Enter the Activation Key .....	18
Step 3: Reboot the System .....	18



Removing ICE.Block from UnixWare/Esix Systems .....	19
Updating ICE.Block on UnixWare/Esix Systems .....	19
Upgrading from Trialware to Licensed Version .....	19

## **Chapter 3: Getting Started ..... 21**

Configuring ICE.Block .....	21
Configuring ICE.Block Using the Windows 95 GUI .....	22
Configuring ICE.Block at the Command Line .....	26
Configuring System Interfaces .....	27
Defining Interfaces Using the GUI Program .....	27
Configuring Interface Names and Device Names.....	29
Defining IP Addresses for Each Interface .....	30
Setting Up Groups .....	32
Setting Up Groups Using the GUI Program .....	32
Setting Up Groups Using Command Line Editing.....	33
Defining Data Packet Filters .....	34
Defining Filters Using the GUI Program .....	35
Defining Filters Using the Command Line Interface .....	38
Using the ICE.Block Control Program .....	41
Activation Key File .....	42
Logging and Reporting .....	43
Data Packet Logging.....	43
Reporting .....	44
E-mail Alerts .....	50



# Introduction

Welcome to ICE.Block™, a secure and affordable UNIX-based Internet *firewall*.

ICE.Block is installed at your point of connection to the Internet, protecting your network from unauthorized access. ICE.Block closes security holes by regulating all data packets passing to and from the Internet. With an easy-to-use, Windows 95 graphical user interface (GUI), you can configure ICE.Block to make your network as accessible or inaccessible as you wish. ICE.Block keeps intruders away from sensitive data and prevents unauthorized access to Internet services, allowing your business to realize the many benefits of the Internet without exposing your network to security threats.

## The ICE Family of Products

ICE.Block is part of the ICE family of products, providing reliable solutions for a variety of connectivity and network security needs:

- ICE.TCP™ connects networked PCs to one or more host computers via TCP/IP. ICE.TCP coexists with local area networks such as Windows for Workgroups, Novell, LAN Manager, Lantastic, and Banyan. In addition to providing its own TCP/IP kernel, ICE.TCP works with other fully-WinSock compatible TCP/IPs, including the Microsoft TCP/IP provided with Windows 95.
- ICE.TEN™ connects individual PCs via serial lines. ICE.TEN provides the same features as ICE.TCP, including a full range of terminal emulation options,



multiple UNIX session capability, bidirectional printing fully integrated with Windows, and easy-to-use file transfer capability.

- ICE.Off.Site™ lets users who are on the road or based in remote locations connect via a modem. ICE.Off.Site delivers the same high-quality, reliable terminal emulation as other ICE products and, when ICE.TEN is installed on the host computer, allows users to take advantage of file transfer and printing features.
- ICE.NFS™ enables PC users running Windows 95 to set up a virtual network drive to access data on a host computer running NFS. Easy to set up and use, ICE.NFS uses Microsoft WinSock TCP/IP and network hardware. ICE.NFS offers several convenient options for mounting file systems and, once a file system is mounted, lets a user access it as if it were a local drive.

J. River, Inc. also provides products and services for connecting to the Internet and remote WANs and LANs. For more information, contact J. River, Inc.:

Telephone: 612-339-2521 • Fax: 612-339-4445

E-mail: [info@jriver.com](mailto:info@jriver.com) • World Wide Web: <http://www.jriver.com>

## **Overview of Network Security**

Security is an important consideration for any organization that relies on computer networks, particularly networks connected to the Internet. A comprehensive security strategy includes policy, procedural, and technical controls that will protect the network from Internet-based threats and regulate the services available to network users.

As a UNIX-based Internet *firewall*, ICE.Block™ is a valuable element in any security strategy, providing a secure yet flexible way to prevent unauthorized traffic between a network and the Internet. But network security is only as strong as its weakest link. We strongly recommend implementing ICE.Block as part of a comprehensive strategy that addresses issues such as access control, information system notices, authorized users and uses, physical access control, password management, and user management.



For example, the ability of ICE.Block to protect your network is seriously compromised if any network user is allowed to establish an individual Internet connection using a modem and PPP (Point-to-Point Protocol). Such *PPP backdoors* make it difficult to control internal Internet access and, more important, may expose the entire network to external security threats from the Internet.

Policies and procedures to prevent PPP backdoors and other Internet security risks should be included in the overall network security policy. Additional information on computer security is available from the Computer Emergency Response Team (CERT) at Carnegie-Mellon University's Software Engineering Institute (accessible via the World Wide Web at <http://www.cert.org>).

## Overview of ICE.Block

ICE.Block™ is a UNIX-based Internet firewall that relies on proven packet filtering technology. Running under SCO, UnixWare, or Esix, ICE.Block protects any IP-based network and is compatible with local area networks such as Novell Netware, Windows (3.x, 95, and NT), and Banyan Vines.

Installed at your point of connection to the Internet, ICE.Block protects your network by using a packet filtering policy that denies all traffic between the network and the Internet that is not expressly permitted. This approach gives the system administrator complete control over who has access to the network and what services authorized users can access. (ICE.Block can also be configured to accept all traffic not expressly denied, a weaker security policy but appropriate for some installations.)

As an *intranet* firewall, ICE.Block also protects sensitive data within your company from unauthorized users and protects individual machines.

ICE.Block controls which protocols and ports are allowed and the direction of traffic — for example, allowing network users to telnet to remote hosts but blocking remote hosts from telnetting to the local network. This provides a high level of security yet gives authorized users access to the Internet capabilities they need.



With ICE.Block, the system administrator can precisely control incoming and outgoing data packets, specifying which are valid and blocking those that are not expressly allowed. ICE.Block correctly handles packets designed to overwrite the headers of previous packets, a commonly-used IP-based attack, and prevents *IP spoofing* attacks, where someone on the Internet pretends to have a local machine's IP address.

Again, we strongly recommend implementing ICE.Block in conjunction with a comprehensive network security strategy that includes appropriate policies and procedures.

## **Requirements and Trialware Agreement**

ICE.Block™ has been developed for and tested on three UNIX operating systems: SCO, UnixWare, and Esix. The specific system requirements are as follows:

### **ICE.Block for SCO**

- SCO 3.2 (Versions 4.0, 4.1, and 4.2)
- SCO Open Server 5
- SCO FastStart

### **ICE.Block for UnixWare**

- UnixWare 2.x

### **ICE.Block for Esix**

- Esix 4.1-1, 4.1-2, and 4.2

ICE.Block will not work with other operating systems and may not work with previous versions of the operating systems listed above.

SCO, UnixWare, and Esix versions of ICE.Block are available. Make sure to use the correct version of ICE.Block for your operating system. Also available for downloading from the J. River Web site (<http://www.jriver.com>) is a Windows 95 GUI firewall configuration and management program for use on PCs.





Once installed on a UNIX computer, ICE.Block operates in the same way as other programs, and is compatible with other programs running on the machine. No other special hardware configuration or installation is required.

**NOTE:** Operating or hosting other applications on the same machine used to host the firewall may compromise network security. It is difficult to control how an application and the operating system interact without placing strict security controls on each. Therefore, the host operating system becomes subject to software bugs and malicious software- or application-based network security attacks. **We strongly recommend that the machine used to host ICE.Block not be used for any other applications.**

Information about new features added since the publication of this manual may be found in UNIX manual pages or the `/etc/iceblock` directory.

ICE.Block is a commercial software product and is the property of J. River, Inc. Your right to use ICE.Block is granted only upon acceptance of the ICE.Block trialware agreement, which limits your use of the software to a trial period of 30 days on a single host computer for purposes of evaluation only. After 30 days, if you choose not to purchase a license for ICE.Block, you agree to remove the software from your system. (Chapter 2, “Installation,” includes instructions for deinstalling ICE.Block.)

**IMPORTANT:** When the 30-day trial period for the trialware version of ICE.Block expires, ICE.Block deactivates. If you do not obtain and enter a permanent activation key (by purchasing the licensed version of ICE.Block), ICE.Block will not protect your network.

## Purchasing a Licensed Version of ICE.Block

The trialware version of ICE.Block provides full functionality for 30 days for evaluation purposes. To upgrade to the permanent, licensed version of ICE.Block, you do not need to reinstall ICE.Block. However, because the activation key for trialware differs from the activation key for the licensed version, you will need to obtain and enter a new, permanent activation key. Without a permanent activation key, ICE.Block will deactivate after the 30-day trial period.



To purchase a licensed copy of ICE.Block, or additional licensed copies for use on other UNIX hosts, contact J. River customer service at 612-339-2521 (Monday through Friday, 8:30 A.M. to 6:00 P.M. Central Time, USA).

## About This Manual

Because of ICE.Block's role as a network security tool, it is important to fully understand its capabilities and operation before using it. System administrators should read through this entire manual carefully before installing and configuring ICE.Block.

Except for minor differences in installation, operating ICE.Block is the same for SCO, UnixWare, and Esix. Wherever differences exist, the manual provides separate procedures for each.

Follow the instructions in Chapter 2 to install ICE.Block.

Chapter 3, "Getting Started," contains information you will need to configure and use ICE.Block.

To alert users to significant issues, the manual provides special *caution* and *warning* notices:

**CAUTION** is used to alert users that the actions described may change the operating characteristics of the firewall, resulting in decreased performance or security.

**WARNING** is used to alert users that the actions described may result in termination of the firewall's operation, leaving the network unprotected by ICE.Block.

## Technical Support

Technical support is available by telephone or E-mail. Before seeking technical support, please review the installation procedure to make sure you have properly



installed and configured ICE.Block. Also check the UNIX manual pages for a discussion of common problems.

For technical support, contact:

Telephone: 612-339-2521 (8:30 A.M. to 6:00 P.M. Central Time, USA)

Fax: 612-339-7056

E-mail: [support@jriver.com](mailto:support@jriver.com)

World Wide Web: <http://www.jriver.com/techsupport>

**IMPORTANT:** For E-mail support, please include the following information:

- Your name
- Address
- Daytime telephone number
- Fax number
- Complete E-mail address
- Version numbers and description of operating environment
- Complete description of the problem

To respond properly, we must have a thorough description of the problem and complete information on how to contact you. If we do not respond to your E-mail within two business days, the E-mail probably did not include a complete E-mail address or telephone number.

## Platinum Technical Support

J. River, Inc. also offers optional *Platinum Technical Support*, which includes free product updates and priority response via a toll-free (U.S. and Canada only) technical support hotline. For more information on Platinum Technical Support, call 612-339-2521 to talk to a J. River, Inc. customer service representative.







## CHAPTER 2

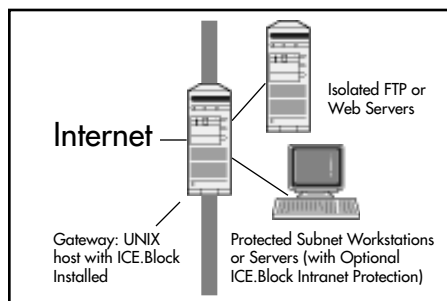
# Installation

INSTALL

## Preparing for Installation

ICE.Block™ is designed and tested for use only with SCO, UnixWare, and Esix operating systems. To help you properly evaluate ICE.Block, the trialware version is *fully functional*: a complete version of the software that expires after 30 days.

Upgrading the trialware version of ICE.Block to the licensed version is easy. When you purchase ICE.Block and register it, J. River, Inc. provides a *permanent activation key* to replace the *temporary activation key* used to install the trialware. You will not need to reinstall ICE.Block.



The UNIX host on which ICE.Block is installed must be the *gateway* to the external network (for example, the Internet) as described in the diagram; otherwise, ICE.Block cannot filter data packets sent to or received from the other machines.

Trialware is downloaded as a compressed image file: iceblock.Z for SCO or iceblkuw.Z for UnixWare. The installation floppy disk produced from the compressed image file will be for *SCO Custom Installation*, *UnixWare Package Installation*, or *Esix Package Installation*. Make sure you obtain the correct image file for your operating system.



The Windows 95 GUI firewall configuration and management program is also available for downloading from the J. River, Inc. Web site at <http://www.jriver.com>. Chapter 3, “Getting Started,” includes instructions for installing and using the GUI program.

## Creating an Installation Disk

If you downloaded the ICE.Block image file to a PC, copy the file to a floppy disk for transfer to the UNIX host where you want to install ICE.Block. Copy the compressed image file from the floppy disk to the tmp directory of the UNIX host by entering the command `doscp -r a:filename tmp` (where *filename* is the name of the image file: **iceblock.Z** for SCO or **iceblkuw.Z** for UnixWare).

Log in to the tmp directory by entering `cd tmp` and decompress the image file by entering `uncompress filename` (again, where *filename* is the name of the image file).

Format a floppy disk by entering `format /dev/rfd0135ds18` then create the installation disk by entering `dd if=iceblock of=/dev/rfd0135ds18`

In addition to the installation disk, you must also obtain a trialware activation key, which is entered during the installation procedure.

## Obtaining a Trialware Activation Key

An *activation key* is required to activate the trialware version of ICE.Block. This *temporary* activation key is valid for 30 days on one Internet host. You will need to enter the trialware activation key during installation to activate ICE.Block.

To obtain a trialware activation key, complete the Trialware Product Registration and Activation Key Request form, located at the J. River, Inc. Web site as <http://www.jriver.com/firewall/firewallform.html>. This form is a simple HTML form that you can complete and submit over the Internet using a forms-compatible Web browser.



If you do not have a forms-compatible Web browser, contact J. River customer service at 612-339-2521 (Monday through Friday, 8:30 A.M. to 6:00 P.M. Central Time, USA).

After completing and submitting the form, you will receive the trialware activation key as a *generation results* report on your Web browser. The activation key consists of two lines of approximately 10 to 12 upper and lower case letters each, for example:

kTImzAefjPBw  
MhxxJpSWcz

**NOTE:** The above is a sample key. Do not use it to activate the trialware version of ICE.Block.

The activation key you receive is valid for one host, as identified in the Trialware Product Registration and Activation Key Request form.



**WARNING: ICE.Block will not work without a valid activation key. If you do not properly enter the activation key during installation, ICE.Block will not activate and your network will not be protected.**

If you are installing the trialware version of ICE.Block on an SCO operating system, follow the instructions in the following section, “Installing ICE.Block on SCO Systems.” To install ICE.Block on UnixWare or Esix, skip to the section “Installing ICE.Block on UnixWare or Esix Systems.” (The installation procedures for UnixWare and Esix are identical.)

The installation procedures also include procedures for deinstalling ICE.Block after the 30-day trial period expires (if you decide not to upgrade from trialware to the permanent, licensed version of ICE.Block).

**INSTALL**



## Installing ICE.Block on SCO Systems

There are no special installation requirements. Install ICE.Block as you would install any SCO component or product.

On SCO Open Server 5 (but not SCO 3.2.x), the *ksl* feature is turned off automatically during installation and reactivated by removal.

### Step 1: Enter System Maintenance Mode

During installation, ICE.Block inserts a module into the system kernel then automatically re-links. Therefore, ICE.Block must be installed while the system is in *system maintenance* (or *single-user*) mode. To do this, reboot the system and enter the *root* password when prompted at boot time.

### Step 2: Install ICE.Block from the Program Disk

Insert the ICE.Block program disk in the floppy drive. (Make sure to use the disk formatted for SCO Custom Installation.)

Execute the Custom utility. From the Custom menus, choose Install Product, then New Product. When prompted by the SCO installer, select the Entire Package for installation.

### Step 3: Re-link the Kernel

If prompted by the installer, enter Y to re-link the kernel.

### Step 4: Enter the Activation Key

After Custom is complete, you must enter the permanent activation key. To enter the permanent activation key, edit the file `/etc/iceblock/iceblock.key` using *vi* or another editor. Be sure to use the exact letter case. Spaces are not allowed. Each of the two sections of the key is on a line by itself, and blank lines are not allowed.





## Step 5: Reboot the System

Finally, to activate ICE.Block, reboot the system using the **reboot** command.

ICE.Block is now installed and active. Until it is configured, however, it will filter all traffic on the network. Follow the instructions in Chapter 3, “Getting Started,” to configure ICE.Block by setting up a configuration file.

**INSTALL**

**CAUTION:** After installation, the firewall will allow no traffic to pass until the configuration file is modified.

## Removing ICE.Block from SCO Systems

To de-install ICE.Block — for example, if you receive an updated version of the software — use the SCO Custom utility. (ICE.Block should be de-installed only while the system is in *system maintenance* — or *single-user* — mode. To do this, reboot the system and enter the *root* password when prompted at boot time.) From the Custom menu, select Remove, then select ICE.Block, then All. After de-installing ICE.Block, the host must be rebooted for the change to take effect.

## Updating ICE.Block on SCO Systems

If you need to update an active ICE.Block firewall — for example, if you receive an updated version of the software — you must first remove the older version by following the instructions in the preceding section. Install the updated software using the standard installation procedure.

**WARNING:** After removal of an older version of ICE.Block, until the new version of ICE.Block is installed and activated, your internal network is not protected: all traffic will be allowed to pass through the host. Disconnecting the internal network at the host will stop inbound and outbound traffic (unless there is another Internet point of entry into the network).



## Installing ICE.Block on UnixWare or Esix Systems

There are no special installation requirements. Install ICE.Block as you would install any UnixWare or Esix component or product.

During installation, ICE.Block modifies the system kernel, which is re-linked when you reboot the system after the installation.

### Step 1: Install ICE.Block from the Program Disk

Insert the ICE.Block program disk in the floppy drive. (Make sure to use the disk formatted for either UnixWare or Esix Package Installation.)

Execute the `pkgadd` installation utility by entering the command `pkgadd -d disketten iceblock` (where `n` is the disk drive where the program disk is inserted, usually 1). For example, for most installations the actual command will be:

```
pkgadd -d diskette1 iceblock
```

### Step 2: Enter the Activation Key

For ICE.Block to work, you must enter a permanent activation key. To enter the permanent activation key, edit the file `/etc/iceblock/iceblock.key` using `vi` or another editor. Be sure to use the exact letter case. Spaces are not allowed. Each of the two sections of the key is on a line by itself, and blank lines are not allowed.

### Step 3: Reboot the System

Finally, to activate ICE.Block, reboot the system using the `shutdown -i6 -g60` command.

ICE.Block is now installed and active. Until it is configured, however, it will filter all traffic on the network. Follow the instructions in Chapter 3, “Getting Started,” to configure ICE.Block by setting up a configuration file.

**CAUTION:** After installation, the firewall will allow no traffic to pass until the configuration file is modified.



## Removing ICE.Block from UnixWare/Esix Systems

To de-install ICE.Block — for example, if you receive an updated version of the software — use the Package Remove utility by entering the command `pkgrm ICEBlock`.

After de-installing ICE.Block, the host must be rebooted for the change to take effect.

**INSTALL**

## Updating ICE.Block on UnixWare/Esix Systems

If you need to update an active ICE.Block firewall — for example, if you receive an updated version of the software — you must first remove the older version by following the instructions in the preceding section. Install the updated software using the standard installation procedure.



**WARNING:** After removal of an older version of ICE.Block, until the new version of ICE.Block is installed and activated, your internal network is not protected: all traffic will be allowed to pass through the host. Disconnecting the internal network at the host will stop inbound and outbound traffic (unless there is another Internet point of entry into the network).

## Upgrading from Trialware to Licensed Version

The trialware version of ICE.Block provides full functionality for 30 days for evaluation purposes. To upgrade to the permanent, licensed version of ICE.Block, you do not need to reinstall ICE.Block. However, because the activation key for trialware differs from the activation key for the licensed version, you will need to obtain and enter a new, permanent activation key. Without a permanent activation key, ICE.Block will deactivate after the 30-day trial period.



**WARNING:** When the trial period (30 days) for the trialware version of ICE.Block expires, ICE.Block deactivates. If you do not obtain and enter a permanent activation key, ICE.Block will not protect your network.

To purchase a licensed copy of ICE.Block, or additional licensed copies for use on other UNIX hosts, contact J. River customer service at 612-339-2521 (Monday through Friday, 8:30 A.M. to 6:00 P.M. Central Time, USA).





## CHAPTER 3

# Getting Started

**GETTING  
STARTED**

This chapter describes how to configure and use ICE.Block. The chapter includes information on defining data packet filters in ICE.Block's configuration file *iceblock.conf* and tailoring ICE.Block's logging and reporting features to your specific needs.

Before configuring data packet filters, make sure a firewall *interface* file is installed and properly configured. For UnixWare users, the interface file is created by the operating system and should not be modified. On SCO and Esix systems, the interface file is created automatically during installation, but can be edited manually when necessary. Refer to the section below, "Configuring the Firewall Interface File."



**WARNING:** If either the interface file or the configuration file is not present, all TCP/IP traffic will pass through the firewall.

## Configuring ICE.Block

The easiest way to configure ICE.Block is through the use of ICE.Block's Windows 95 GUI, a configuration and management program that lets you set up the interface file, define data packet filters, and set up a logging and reporting process. The Windows 95 GUI (available for downloading from the J. River Web site at <http://www.jriver.com>) should be installed on a PC connected to the protected network address space.

**IMPORTANT:** To use the GUI program, you must first *flush* (remove) the filters set up by default during installation. (No traffic can pass through the firewall until



it is configured.) To flush existing filters, enter the following command on the firewall computer:

```
/etc/iceblock/iceblock -f
```

You also have the option of editing ICE.Block's interface and configuration files and program command lines directly. Command line editing may be preferable for users who desire more precise control over packet filtering, logging, and reporting functions. It can also be helpful for debugging and troubleshooting.

**CAUTION:** If you use the command line interface, later use of the Windows 95 GUI will probably result in the loss of data packet filter rules and other configuration information defined using the command line interface.

## Configuring ICE.Block Using the Windows 95 GUI

The GUI configuration and management program for Windows 95 is downloaded as a self-extracting compressed file. To install it, click on the Start button in Windows 95, then click Run to display the Run dialog box. In the Open field enter the filename and click OK to run the installation program. Follow the on-screen instructions to complete the installation.

The GUI program is installed in the Program Files folder in a subfolder called ICE.Block. Use the Start button to open the folders and run the ICE.Block program shortcut. (ICE.Block is pre-configured to allow communication between the Windows 95 GUI on the PC and the firewall. However, you may need to flush (remove) firewall filters to let traffic pass. This is done with the command `/etc/iceblock/iceblock -f` on the firewall machine.)

When the GUI program runs, it displays buttons that let you switch between six main windows: Interface, Groups, Services, Logging, Write, and Lock. (The Write and Lock windows are used only for saving your configuration information and locking the firewall after the configuration session, as explained later.)

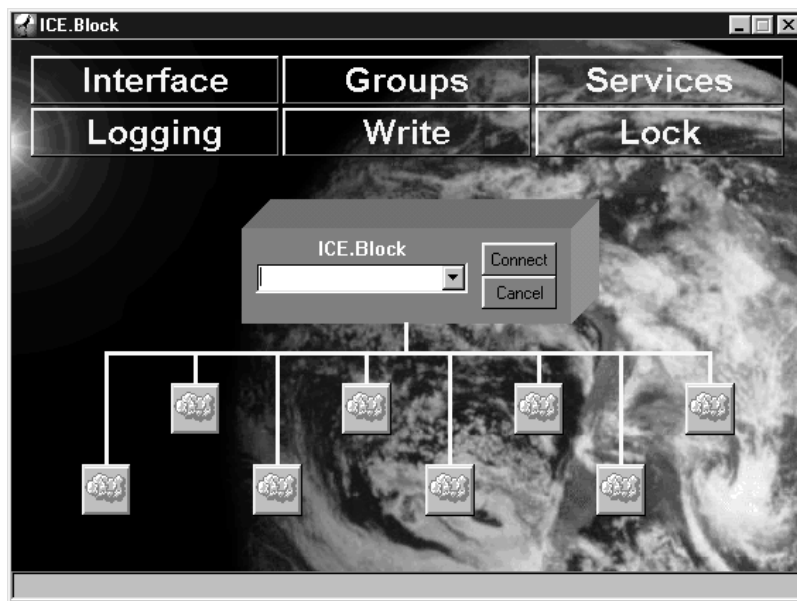


Figure 3.1: Windows 95 GUI Configuration and Management Program

At the top of each window is a standard Windows 95 title bar. Clicking on the skater icon on the left side of the title bar displays an Options menu that lets you perform standard Windows functions: Restore, Move, Minimize, and Close. You also can use the Options menu to display program information or toggle to and off the background image displayed in the window.



Figure 3.2: Options Menu

The right side of the title bar displays shortcut buttons to Minimize the window or Exit the program. (Note: The Windows 95 Maximize/Restore button is not functional.) The program will always prompt you to save changes before exiting.



## Defining a Firewall Host

By default, the GUI program displays the Interface window when it initially runs. The first time you use the GUI program, you must define the name of the UNIX host where ICE.Block is installed. In the box labeled ICE.Block, enter the name of host and click the Connect button. For subsequent sessions, the name will be saved and displayed in a pull-down menu. (A name can be deleted by selecting it from the pull-down menu then pressing the DEL key.)

The program will prompt you for a password. Enter the root password of the firewall host and click OK. Expect a slight delay as the password is checked. (The password is encrypted before being sent across the network to prevent snoopers from seeing it.)

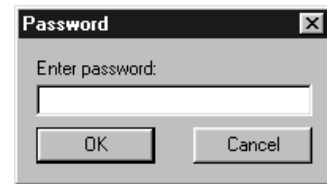


Figure 3.3: Password Dialog Box

**CAUTION:** Because the password is sent across the network without a Central Signature Authority, it is susceptible to “man-in-the-middle” attacks. Don’t use the Windows 95 GUI unless all systems in the network between PC and the firewall host are trustworthy.

After clicking the Connect button to establish communication with the firewall host, you can cancel the process (and stop the parsing of firewall data) at any time by clicking the Cancel button.

If the Windows 95 GUI program reports any errors during start-up, the program will load default parameters. Any configuration information previously saved will be completely overwritten when the GUI program saves new information. If no errors are reported during start-up, the GUI program will load existing configuration information.

The following sections in this chapter explain how to configure ICE.Block using the Interface, Groups, Services, and Logging windows in the GUI program.





## Saving Configuration Information and Locking the Firewall

After configuring ICE.Block, you must save the configuration information and lock the firewall.

To save configuration information, click on the Write button, which displays a warning prompt, then click on the Save button.

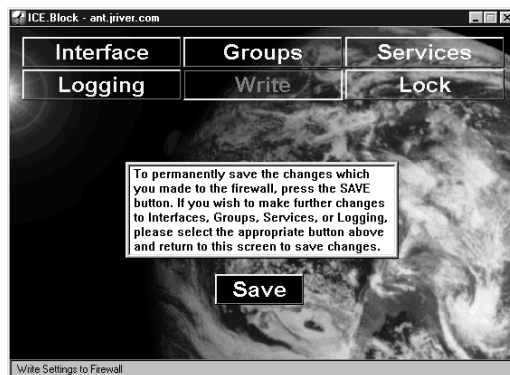


Figure 3.4: Write Window

Figure 3.5: Lock Window



After saving the configuration information, you can lock the firewall. As noted earlier, ICE.Block is pre-configured to allow the Windows 95 GUI access to the firewall host. To prevent an attacker from accessing ICE.Block from a remote PC using the GUI program, click on the Lock button, which displays a warning prompt, then click on the second Lock button.

You also can lock the firewall manually by editing the `/etc/inetd.conf` file on the UNIX host to remove the `iceguid` command line.

Access to the firewall via the Windows 95 GUI can be restored for later configuration sessions by executing the command `/etc/iceblock/unlock` on the UNIX host.



**CAUTION:** If access to the firewall via the Windows 95 GUI program is not blocked (using the `Lock` command), attackers may be able to enter the system and remove packet filters.

## Configuring ICE.Block at the Command Line

While an in-depth knowledge of UNIX is not required to configure and manage ICE.Block, command line editing of ICE.Block's interface, configuration, and program files, particularly the configuration file `iceblock.conf`, can provide deeper insight into how ICE.Block works and give users greater control over ICE.Block functions. Configuring ICE.Block using the command line interface can also provide greater security.

Information about all commands in the files is included in corresponding UNIX manual pages, which you can access with the `man` command. For example, to access manual pages for `iceblock` (which includes a list of all the manual pages available), enter `man iceblock`.

ICE.Block is pre-configured to allow access to the firewall using the Windows 95 GUI (as described above). If you plan to rely exclusively on the command line interface for configuring ICE.Block, “lock out” the Windows 95 GUI program by deleting the `iceguid` line from the `/etc/inetd.conf` file.

**CAUTION:** If you do not “lock out” the Windows 95 GUI program by deleting the command line `iceguid` from the `/etc/inetd.conf` file, an attacker may be able to enter the system and turn off the firewall from a remote PC using the Windows 95 program.

The operation of ICE.Block's programs can be modified by adding command options to program command lines. The system interface file and configuration file are UNIX text files that can be edited using a text editor such as `vi` or `emacs`. Blank lines, which are allowed, are ignored. Comments, which begin with the pound (`#`) symbol, are also ignored.



## Configuring System Interfaces

Essential to the operation of ICE.Block, the interface file maps the names of interfaces used in the iceblock.conf configuration file to the actual devices used for network traffic (either network interface cards or PPP connections). Without a properly configured interface file, the data packet filters you define in iceblock.conf will have no effect on the system.

In UnixWare installations, the interface file is installed and configured automatically as /etc/confnet.d/inet/interface and must not be modified.

In SCO and Esix installations, the interface file (/etc/iceblock/interface) is created automatically during installation. If a network card is added or removed, however, the interface file must be modified manually to reflect the change.

There are two kinds of interfaces: *network card* and *PPP*. The number of network card interfaces supported by ICE.Block is limited only by the number of network cards supported by the operating system (although only eight interfaces can be configured using the Windows 95 GUI program). ICE.Block supports one PPP interface (with the potential for multiple connections).

Each interface must specify an *interface name* and a *device name* (as used by the operating system). These are defined in the interface file.

In addition, you have the option of specifying IP addresses associated with the system (to prevent IP spoofing attacks) and a name of your own choosing that is easier to remember than the system name (available only if you are using the Windows 95 GUI program).

### Defining Interfaces Using the GUI Program

In the Windows 95 GUI program, interfaces are configured from the Interface window. Interfaces are represented by cloud icons located below the ICE.Block host icon. The Windows 95 GUI program allows you to configure a maximum of eight interfaces.

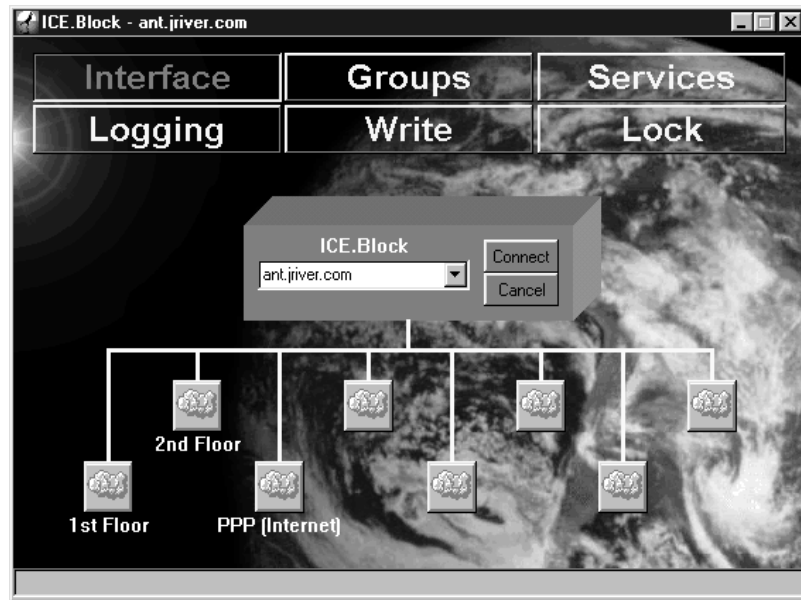


Figure 3.6: Interface Configuration Window

When you pass the cursor arrow across an icon, the arrow will change to either a pointer finger, which indicates the interface can be edited, or an X, which means it cannot.

Clicking on an icon with the pointer finger displays an Interface Information dialog box. In addition to letting you define an interface name, device name, and network IP addresses, the Interface Information box lets you specify a name for the interface that's easier to remember. This name will be displayed under the interface's icon in the main Interface window. (See the next section for information on interface names and device names.)

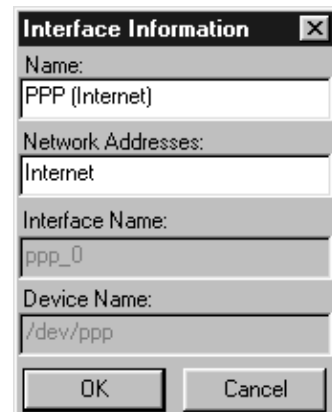


Figure 3.7: Interface Information Dialog



After defining information for an interface (see following sections), click OK to save the settings and return to the main Interface window. Click Cancel to exit the Interface Information box without saving settings.

## Configuring Interface Names and Device Names

In UnixWare, the interface and device names are defined in the interface file automatically by the operating system and should not be changed.

GETTING  
STARTED

For SCO and Esix installations, the interface file is created during installation. If necessary, the file may be edited to change the interface and device names.

The format for each interface is:

**name:unit:unused:device:unused...**

where:

**name** is the interface name,

**unit** is a unit number appended to the interface name, and

**device** is the device name.

To find the interface and device names for a network card, look in the `/etc/strcf` file. Slightly below the line `boot()`, look for an uncommented line (no `#` symbol at the beginning) similar to the following:

```
cenet ip /dev/smpw0_ smpw0 0
```

Using the above example, the name to enter in the interface file for the network card is `smpw0`, the unit is `0`, and the device is `/dev/smpw0_0`.

The interface name for PPP is always `ppp_0`. This must not appear in the interface file.



In UnixWare, the network card parameters appear in the first, second, and fourth fields of the interface file. They should be changed only by adding or removing network cards. Do not edit the interface file manually!

Interfaces must have the icemod module pushed on them. In UnixWare, this is handled by the `/etc/ap/chan.ap` file, which is modified automatically during installation. Whenever the networking hardware is changed (for example, by adding or removing cards), this file must also be changed. In SCO, this is handled by the `strcf` file, and happens automatically.

## Defining IP Addresses for Each Interface

A range of IP addresses (also referred to as network addresses) can be defined for each interface. Defining IP addresses is optional, but should be performed to prevent IP spoofing attacks. See “Rules for IP Addresses” below for the acceptable formats for IP addresses.

If you are defining IP addresses for interfaces using the GUI program, enter a list of IP addresses in the Network Addresses field of the Interface Information dialog box. The members of the list must be separated by blanks.

If you are defining IP addresses using the command line interface, edit the `iceblock.conf` file. The `iceblock.conf` file contains two sections: the first controlling the valid IP addresses for each interface, and the second describing data packet filters (discussed later). The interface, or *network*, section must come first.

The interface section of `iceblock.conf` consists of a series of network lines (one for each interface), each of which uses the following syntax:

**network *name* *IP-addresses***

where:

***name*** is the interface name of the network, created by combining the first field, an underscore (`_`), and the second field from the corresponding line of the interface file. (For example, if the line from the interface file is `NE2000:0::/dev/NE2000_0`, the interface name is `NE2000_0`.)



*IP-addresses* is a range of addresses.

Each network line restricts the source IP address of incoming packets for that network. IP addresses can be specified in several ways, as discussed in the next section.

The IP address for the last network line also can be defined as **internet**; which means any address not previously listed. A network line with the **internet** keyword means accept only incoming packets on that interface that do not have a source address in any local network. This prevents IP spoofing attacks, where someone on the Internet pretends to have a local machine's IP address. It also prevents local users from engaging in IP spoofing on the Internet.

For example, let's say the firewall host has two local networks, named NE2000\_0 and NE2000\_1, and a PPP connection to the Internet, named ppp\_0. Host 1 is on the first local network (Class C), and host 2 is on the second local network (Class C with 3 additional subnet bits). The following prevents IP spoofing either from the Internet to some local machine, or from any local machine to the Internet (or the other local network).

```
network NE2000_0 host1/24
network NE2000_1 host2/27
network ppp_0 internet
```

## Rules for IP Addresses

IP addresses can be specified as an asterisk (\*), which includes everything. When occurring as a source or destination address in a rule, \* includes everything except spoofed addresses. Other formats are: an IP address, as dotted notation or *fully qualified domain name (FQDN)*; or an IP address appended with a slash (/) followed by a number of bits. For example, a Class C network can be described as 195.145.155.132/24 (the IP address with 24 bits of host address).

IP addresses may also be a group name from the group file. See the following section for more information on setting up groups.



Here are some other examples:

195.145.155.132	A single machine with the IP address 195.145.155.132.
195.145.155.132/24	Any machine on the same Class C network.
195.145.155.0/24	Same as above.
195.145.155.132/27	Any machine on the same subnet of the Class C network with three bits of subnetting.
195.145.155.0/27	Not the same as above, since 0 is on a different subnet than 132.

For the above examples, a valid host name (FQDN) could be substituted for the IP address to produce the same results. For example, 195.145.155.132/24 might also be expressed as machine.jriver.com/24.

## Setting Up Groups

When you configure ICE.Block, you can streamline the configuration process by setting up user groups. A group is simply one or more hostnames or IP addresses organized under any name you choose. Organizing your network of users into groups can provide a more convenient way of defining data packet filters.

For example, you can set up user groups according to organization function — i.e., marketing, sales, administration, etc. — then define data packet filters that provide an appropriate level of Internet access for each group.

### Setting Up Groups Using the GUI Program

To set up groups with GUI program, click the Groups button to display the Groups window. Group names are listed in the Group Name box (to the left). To add a new group, click the Add button below the Group Name box, enter the name of the group, and hit ENTER. To remove an existing group, click on the group to highlight it, then click the Remove button.

After creating a new group name, define the group using the Group Definition box (to the right). After clicking on a group name to highlight it, you can define groups by either hostname or IP address. To enter a new hostname/IP address,



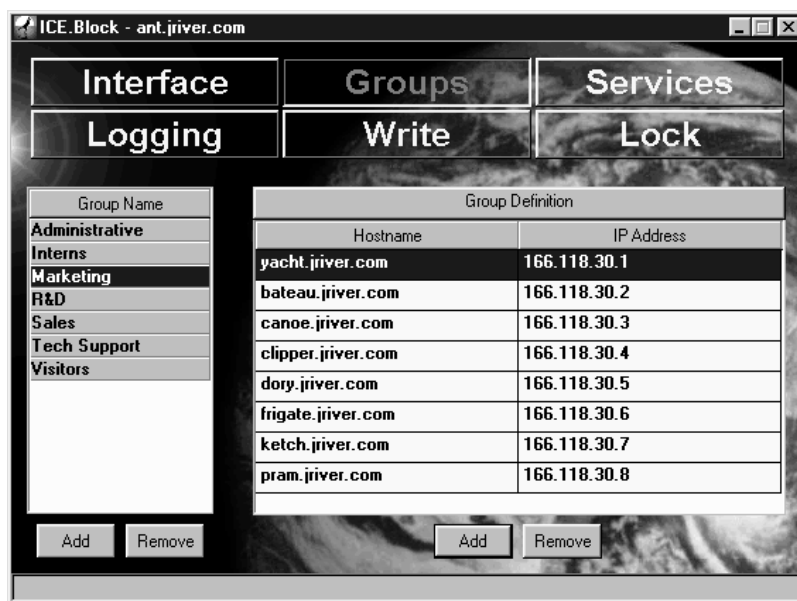


Figure 3.8: Groups Window

GETTING  
STARTED

click the Add button below the Group Definition box and enter a hostname. Or double-click in the IP address box to enter an IP address. When you finish entering the new hostname or IP address, hit ENTER.

To remove an existing hostname/IP address, click to highlight it, then click the Remove button.

After you set up your groups, the group names you selected will appear as options under the Source and Destination designations in the Services window (where data packet filters are defined). You can now define data packet filters by group, as described in the next section.

## Setting Up Groups Using Command Line Editing

To set up groups using the command line interface, edit the file `/etc/iceblock/iceblock.group`. Each line describes one group. The group name comes first, followed by a colon (:), followed by the members of the group



separated by colons. For example, if group admin had machines a, b, and c, the line would be:

```
admin:a:b:c
```

## Defining Data Packet Filters

By default, ICE.Block will reject all traffic to and from the Internet that is not expressly permitted. (The default can be reversed: to accept all traffic not expressly denied.) This permission comes in the form of data packet filters defined in the configuration file `iceblock.conf`. The `iceblock.conf` file controls firewall behavior when the **iceblock -i** command is issued (usually when the machine boots).

ICE.Block can filter packets using the *Transport Control Protocol (TCP)*, *User Datagram Protocol (UDP)*, and *Internet Control Message Protocol (ICMP)*. The most common is TCP, which is used in services such as *File Transfer Protocol (FTP)* and *Telnet*. UDP is used in *Domain Name Services (DNS)* and *Network File Services (NFS)*. ICMP is used for *ping* and other network control functions. In addition, ICE.Block uses a subclass of TCP, named *tcp/est*, for packets that are part of an already-established session.

ICE.Block can filter any service running on TCP, UDP, or ICMP, including:

### **rlogin and telnet**

*Remote Login* and *Telephone Net* are login services. They permit remote users to login on your system.

### **finger, netstat, whois, systat, and who**

These are system status services from which hackers can glean valuable user, network, and system process information.

### **printer**

This is a system resource service that allows other computers to run print jobs. This service can be used to copy files to and from your system.

**ftp**

FTP allows the transfer of user and directory files.

**rpc**

*Remote Procedure Call* is the basis for all NFS services.

**bootp and tftp**

These are diskless-workstation services that allow booting and file transfer.

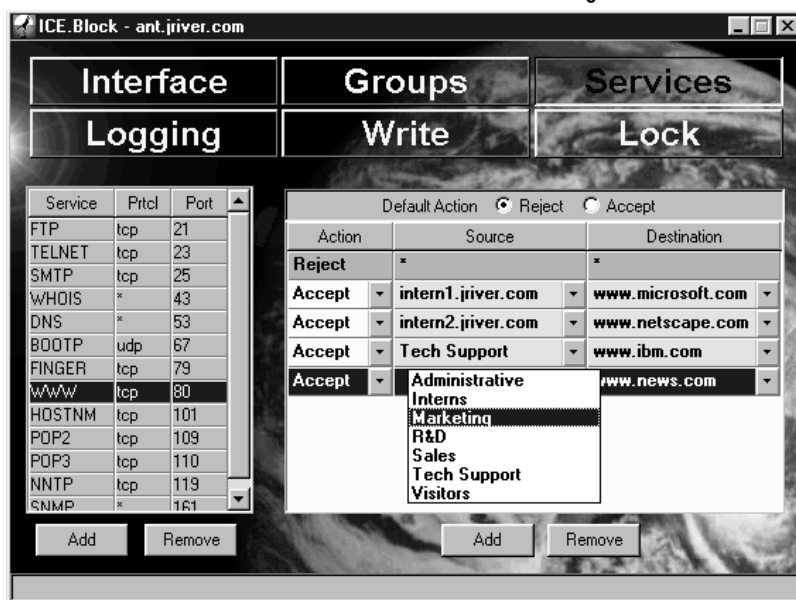
**GETTING  
STARTED**

## Defining Filters Using the GUI Program

To define data packet filters using the Windows 95 GUI program, click the Services button to display the Services window. (The term services refers to the various types of data packets that pass to and from your network.)

A list of services (with the corresponding protocol and port for each) is displayed to the left. All common services are predefined.

Figure 3.9: Services Window





To add a new service, click the Add button, which displays a Service Information dialog box where you can enter the name of the service, the protocol (**icmp**, **udp**, **tcp** or **tcp/est**), and port number. After entering service information, click OK. Each new service is added to the end of the list.

Figure 3.10: Service Information Window

To delete a service listing, click on the service to highlight it, then click Remove.

The box to the right defines filtering information for each service. The Default Action radiobuttons at the top of the box set the default (either reject or accept) for all services. For example, if the default is reject, the firewall will reject data packets for all services until you specifically define which data packets will be accepted (based on source and destination).

Figure 3.11: Services Window

Service	Prctl	Port
FTP	tcp	21
TELNET	tcp	23
SMTP	tcp	25
WHOIS	*	43
DNS	*	53
BOOTP	udp	67
FINGER	tcp	79
WWW	tcp	80
HOSTNM	tcp	101
POP2	tcp	109
POP3	tcp	110
NNTP	tcp	119
SNMP	*	161

Default Action <input checked="" type="radio"/> Reject <input type="radio"/> Accept		
Action	Source	Destination
Reject	*	*
Accept	intern1.jriver.com	www.microsoft.com
Accept	intern2.jriver.com	www.netscape.com
Accept	Tech Support	www.ibm.com
Accept	Administrative	www.news.com
	Interns	
	Marketing	
	R&D	
	Sales	
	Tech Support	
	Visitors	



**CAUTION:** If you change the default filter setting (i.e., reject or accept), the new default is applied to ALL services, not just the service currently highlighted.

To define filtering information for a service, click on the service (in the Service box to the left) to highlight it. In the filtering information box to the right, set action (either accept or reject), source, and destination.

For source and destination, you can use an asterisk (\*), hostname, IP address, or group name (if groups have been defined, as described in the previous section). An asterisk (\*) allows all possible values.

Three services deserve special mention: ICE.Block, FTP, and FTP PASV.

ICE.Block (port 969, TCP) is the port used by the GUI configuration program to communicate with the firewall host computer. It must be enabled for any Windows 95 machines that will be used to run the GUI program.

FTP allows FTP using *dynamic packet filtering*. In the Service Information box for FTP, the Enable Dynamic FTP checkbox must be activated if FTP is used. With this, FTP data packets are passed only if they come from a machine with an existing connection through the firewall.

FTP PASV is a more secure way of using FTP. If the FTP client and server both allow PASV mode, then the client is responsible for starting all connections.

Both FTP protocols, dynamic and PASV, have security holes. With dynamic filtering, an attacker on the FTP server can take advantage of open FTP connections to break through the firewall. If FTP PASV is used to allow local machines FTP access to the Internet, outsiders are prevented from breaking in, but local users can attack the Internet on ports larger than 1024. If FTP PASV is used to allow Internet users access to a local machine, they are allowed access from any port larger than 1024 to any port larger than 1024.



**NOTE:** There is a third way to handle FTP: allow access from port 21 on the server to any port greater than 1024 on the client. Because this opens up all ports larger than 1024 for any machine with FTP privileges on the local network, it is not supported by the GUI.

Refer to the next section for more detailed information on configuring data packet filters.

## Defining Filters Using the Command Line Interface

The configuration file `iceblock.conf` is a UNIX text file that can be edited using a text editor such as *vi* or *emacs*. Blank lines, which are allowed, are ignored. Comments, which begin with the pound (#) symbol, are also ignored.

The `iceblock.conf` file contains two sections: the first controlling the valid IP addresses for each interface (as described earlier), and the second describing data packet filters.

The filtering section of the `iceblock.conf` file must follow the interface (or *network*) section. Each line in the `iceblock.conf` file describes one group of packets to be accepted or rejected. When the firewall receives a packet, it searches each line in the `iceblock.conf` file from the bottom up for the first matching line, which determines whether the packet is accepted or rejected. If a matching line is not found the packet is rejected.

The syntax for filters is:

***action int dir prot srcip srcport dstip dstport***

***action*** is **accept** or **reject**.

***int*** is the *interface*, as specified in the interface file.

***dir*** is the *direction*, either **upstream** or **downstream**. These refer to the direction the packet travels in relation to the firewall: upstream is from the



network to the firewall, and downstream is from the firewall to the network. They have nothing to do with whether the packet comes from the Internet, or is going to the Internet.

**prot** is the *protocol*, either **icmp**, **udp**, **tcp** or **tcp/est**. The tcp/est protocol is a way of describing tcp packets that are part of an already-established connection; that is, any tcp/est packet is also tcp, but some tcp packets (the ones that create the connection) are not tcp/est. The conventional approach is to reject packets only for tcp, and accept all tcp/est packets.

**srcip** is the *source IP address* and may be specified numerically, as a hostname, or as a group name (as described earlier in the “Rules for IP Addresses” section). As an option, it can be followed by **/number**, where **number** is the number of significant bits. For example, 195.145.155.161 and jriver.jriver.com each specify a single host (the same one), but 195.145.155.161/24, 195.145.155.0/24, and jriver.jriver.com/24 each specify the complete Class C address range 195.145.155.0 to 195.145.155.255.

**dstip** has the same format as **srcip**, but describes the *destination IP address* of the packet.

**srcport** and **dstport** describe the *source port* and *destination port* of the packet. These can be either numbers (in the range 0 to 65535) or symbolic names (from the file /etc/services). The prefixes <, >, <=, or >= can be used to specify ranges. For example, >1024 specifies all non-privileged ports (ports greater than 1024).

An asterisk (\*) can be used in any field except *action* to allow all possible values. ICMP may have no valid value in the *port* fields except for \*.

## EXAMPLE

Let's say our host, fw, connects to DNS server dnshost. The server has two interfaces: PPP to the Internet and NE2000 to our local network. We want to allow incoming ftp services from all hosts, except badguys.evil.com. We must



allow routing information to be passed, and we decide to allow all outgoing services. The following is an example of an appropriate iceblock.conf file:

```
network NE2000_0 fw/24
network ppp_0 internet

accept ** icmp * * * * # accept all icmp packets

accept * * * fw * dnshost domain
accept * * * fw domain dnshost *
accept * * * dnshost * fw domain
accept * * * dnshost domain fw *
accept * * * route * route # routing tables

accept ** tcp * * * ftp # accept incoming ftp connections
# reject everything from bad guys, including ftp
reject * * * badguys.evil.com * * *

accept * * * fw/24 * * * # allow all outgoing connections.

# accept any already established connection
accept ** tcp/est * * * *
```

These rules place the same filters on all interfaces. To be more specific, let's filter incoming telnets (from the Internet) and outgoing http (from the local net). In addition, to make the filtering occur as quickly as possible (which eliminates the slight chance of corrupting the firewall), add the following lines to the iceblock.conf file:

```
reject ppp_0 upstream tcp * * * telnet
reject NE2000_0 upstream tcp * * * http
```





## Using the ICE.Block Control Program

The *iceblock* program, installed in the */etc/iceblock* directory, controls the ICE.Block firewall, installing and removing filters, verifying the activation key, and controlling console logging.

When ICE.Block is installed, it configures your system to run the *iceblock* program during system start-up, but you also can run *iceblock* from the shell prompt to debug configuration files. To run the *iceblock* program and change the state of the firewall, the correct activation key must be entered in the file */etc/iceblock.key*.

GETTING  
STARTED



**WARNING:** If an error exists in the configuration or key file, ICE.Block will not function. An error message will be displayed during start-up, but may scroll off the screen quickly.

Entering the command **iceblock** (with no command options) will display the program version number. Several command options can be added to the *iceblock* command line to direct the program to perform special functions. The syntax for *iceblock* command options is:

*/etc/iceblock/iceblock -ifpcCkvdD file*

**-i** This command option installs the data packet filters defined in the *iceblock.conf* configuration file. To run the *iceblock* program with a different configuration file, the name of the file should be entered at the end of the command line. If the activation key is not valid or the *iceblock.conf* contains errors, no filters will be installed. Any previously installed filters are removed first.

**-f** This option flushes the filters currently installed, disabling the firewall by allowing all data packets to pass.

**-p** This option parses the filters in the *iceblock.conf* file. To parse filters in a different configuration file, enter the name of the file at the end of the command line.



**-c** Use this option to enable console logging. (Use the command option **-C** to disable console logging.) When console logging is enabled, a message is sent to the console for each packet filtered. Messages are displayed in the following format:

**NOTICE: Rejected cnt, int dir prot source->dest**

where **cnt** is the number of packets like this filtered since the iceblock daemon last gathered statistics, **int** is the major number of the device where the packet was filtered (which shows up as a minor number since network devices are cloning devices), **prot** is the protocol of the packet, and **source** and **dest** are the source and destination IP addresses of the packet.

**-C** This option disables console logging (see option **-c** above), preventing packet reject messages from appearing on the console.

**-k** Entering the iceblock command with this option verifies the activation key (see “Activation Key File” below).

**-v** Entering the iceblock command with this option displays a version message.

**-d** This command option enables dynamic packet filtering for FTP.

**-D** This command option disables dynamic packet filtering for FTP.

To make the settings for dynamic packet filtering and console logging permanent, edit the file `/etc/iceblock/iceblock.rc` to change the definitions at the top of the file. Otherwise, the change remains in effect only until the next reboot.

## Activation Key File

The key file, installed as `/etc/iceblock/iceblock.key`, contains the activation key used to unlock the firewall. The key file consists of two lines of approximately 10 to 12 upper and lower case letters each, for example:



```
kTImzAefjPBw  
MhkxJpSWcz
```

For ICE.Block to work, these lines must be a valid key from J. River, Inc. The activation key is obtained by completing the Product Registration and Activation Key Request form, which J. River uses to generate an activation key for your specific firewall host. If the key file does not exist or the key lines are not valid, the firewall cannot be used.

GETTING  
STARTED

## Logging and Reporting

ICE.Block offers a set of highly-configurable logging and reporting features that enable a system administrator to precisely monitor attempts to break through the firewall. Data packets filtered out by the firewall can be sorted and provided as regular reports. ICE.Block can also be configured to send immediate E-mail alerts when certain events occur.

The logging and reporting system consists of three programs: *iceblockd*, the data packet logging daemon; *icelog*, the reporting program; and *icewatch*, the E-mail alert program.

Logging and reporting features are easily configured using the Windows 95 GUI firewall management program or command line editing.

### Data Packet Logging

Information about data packets filtered by the firewall is retrieved and logged by the *iceblockd* daemon, located in the `/etc/iceblock` directory. It communicates with the kernel periodically, retrieving a list of packets filtered and new rules installed (if any).

By default, *iceblockd* runs during system start-up (via the script `/etc/iceblock/iceblock.rc`), but you also can run *iceblockd* from the shell prompt to debug configuration files. (Be sure to kill previously-running instances before starting a new instance.)



By default, iceblockd collects information every 15 seconds and writes the information into log files located in the `/etc/iceblock/log` directory. Error information is also collected and written to the file `/etc/iceblock/iceblockd.log`.

Log filenames are formatted according to the date and time of collection: `YYYYMMDDhhmmss` (where `YYYY` is the year, `MM` is the month, `DD` is the day, `hh` is hour, `mm` is minutes, and `ss` is seconds). For example, the filename `19961010123035` is assigned to information collected on October 10, 1996 at 12:30:35 P.M. A new log file is created each time filters are installed (normally once each time the system boots).

The default settings for frequency of collection and location of log files should work well for most users, but you can change these settings easily by adding command options to the `iceblockd` command line. The syntax is:

**`iceblockd -tseconds -ddirectory`**

**`-tseconds`**

where ***seconds*** is the number of seconds between information requests. Smaller values increase the load on the system; larger values increase the possibility of losing log information (and E-mail warnings via the `icewatch` program). Note that changing this value requires a corresponding change in the `icewatch` program (see the “E-mail Warnings” section).

**`-ddirectory`**

where ***directory*** is the name of the directory in which log files will be stored.

## Reporting

Information collected and logged by the `iceblockd` daemon and the rules used for filtering data packets can be organized into regular reports using the `icelog` program.

By default, `icelog` is run once a day by `cron`, but you also can run it manually. (The log files are not removed, and must be removed manually to recover disk



space.) Removing the last log file causes the iceblockd daemon to lose data until the command **iceblock-i** is executed or the system rebooted.

By default, icelog reports the rules and data packets filtered from the previous day, sorted by the source IP address.

Sorting and reporting parameters are easily defined using the Windows 95 GUI program or command line editing.

**GETTING  
STARTED**

### Configuring Reporting Using the GUI Program

To set up sorting and reporting parameters using the Windows 95 GUI program, click the Logging button to display the Logging window. In the top half of the Logging window is a sub-window controlled by three tabs that let you toggle between dialog boxes in which you can set up reports, set up E-mail alerts, and activate console logging. (E-mail alerts and console logging are described later in the section “E-mail Alerts.”)

Figure 3.12: Logging Window with Reporting Tab Selected

ICE.Block - ant.jriver.com

Interface Groups Services  
Logging Write Lock

Reporting Alert Console

Grouping interval for reporting violations: 5 minutes.

From: ftp.cdrom.com To: ftp.jriver.com Port: 21

Sort by: Source E-mail: report@jriver.com

Add Edit Remove

Type	From	To	Port	Sort	Thresh	E-mail
Report	*	secret.jriver.com	80	Source	5	report@jriver.com
Report	ftp.cdrom.com	ftp.jriver.com	21	Source	5	report@jriver.com

Configure Logging



Each report you set up is displayed in the lower half of the Logging window.

To set up a new report, click on the tab labeled Reporting. In the dialog box, enter the parameters for sorting data packets filtered out by the firewall, including a time interval (or *threshold*) for grouping violations; the *source*, *destination*, and *port* from which violations are collected; the order in which the violations are *sorted* (i.e., by source, destination, or port); and an *E-mail address* where reports should be sent.

Figure 3.13: Dialog Box Displayed Using Reporting Tab

After entering the report parameters, click the Add button to add the new report entry to the list.

To change the parameters of an existing report entry, click in the listing to highlight it, click the Edit button to display it in the dialog box, make the changes, and click Submit.

To delete an existing report entry, click in the listing to highlight it, then click the Remove button.

Refer to the next section for more detailed information on reporting parameters, as defined using command line editing.

### Configuring Reporting Using Command Line Editing

At the command line, settings are changed by adding command options to the `icelog` command line. The syntax is:

```
icelog -nastp -Ssource -Ttarget -Pport -bbegin -eend -iinterval -ddir
```



**-n** Don't report the internal version of the rules used.

**-a** Report all packet information available.

**-s** Sort by source address first (the default).

**-t** Sort by target address first.

**-p** Sort by target port first.

**-Ssource**

Restrict reporting to packets from the *source address*, as defined in the `iceblock.conf` configuration file. The default is all sources.

**-Ttarget**

Restrict reporting to packets that go to the *destination address target*, as defined in the `iceblock.conf` configuration file. The default is all targets.

**-Pport**

Restrict reporting to packets that go to the *destination port*, as defined in the `iceblock.conf` configuration file.

**-bbegin**

Set the time at which the report begins. This parameter must be in the same format as log file names (YYYYMMDDhhmmss, see “Data Packet Logging” above). The report will include only packets filtered after this time. The default is the most recent midnight minus 24 hours.

**-eend**

Set the time at which the report ends. This parameter must be in the same format as log file names (YYYYMMDDhhmmss, see “Data Packet Logging” above). The report will include only packets filtered before this time. The default is the most recent midnight.

***-iinterval***

This command option defines the *time interval* in which information is summarized. The default is 15 seconds. Specifying a smaller value than the time interval specified for the logging daemon iceblockd has no effect.

***-ddir***

Defines the *directory* where log files are collected for reporting. The directory should be the same directory used by the logging daemon iceblockd. For example, if you change the directory where iceblockd places log files, make the same change here.

**Reporting Output**

Reporting output is formatted as the rules in use followed by the number of data packets filtered according to those rules, for as many rule sets as were in use during the time period specified. If the *-n* option is specified (see the preceding section), rules are not included in the output.

The rules represent the data structure used by the kernel. While the structure is not in the same format as the configuration file *iceblock.conf*, it can be useful in debugging rules during the configuration process. The output includes a header telling when the rules were installed followed by the rules themselves.

Rules are partitioned into groups according to interface, direction, and protocol. For example, if two interfaces (such as a network card and PPP) exist on the firewall machine, the output will include 16 sections (two interfaces, each with two directions and four protocols). Each section describes the packets accepted (by source and destination ports and source and destination IP addresses). The output will include several lines describing the set of ports:

Ports: <range> -> <range>

Following each of these will be several lines describing a range of IP addresses:

<range> -> <range>





Rules for the ICMP protocol do not include lines for ports, since ICMP does not use ports.

Here's an example:

NE2000 interface, downstream direction, tcp protocol

```
Ports * -> 0...6
* -> *
Ports * -> 7
jriver -> test
Ports * -> 8...65535
* -> *
```

GETTING  
STARTED

The NE2000 interface, in this example, is the local network. The rules displayed apply to tcp packets in the downstream direction. The rules specify that packets are accepted for any source port, and any destination port smaller than 7. For any source port, if the destination is 7, the packet is accepted only if the source machine is jriver and the destination machine is test. For any source port, and any destination port larger than 7, the packet is accepted.

These might be generated from the file:

```
accept NE2000 downstream tcp * * * *
reject NE2000 downstream tcp * * * echo
accept NE2000 downstream tcp jriver * test echo
```

Note that in actual use the names jriver and test should be fully qualified (e.g., jriver.jriver.com and test.jriver.com).

Following each rules section are the packets filtered by that section, summarized by intervals of 15 seconds, or the time specified by the -t option. For example:



Fragmentation Overwrite Attacks 14:35:45

1: 1.1.1.1(1096) -> jriver(smtp), ppp, upstream, tcp

Filtered Packets 14:35:45

2: elephant(1046) -> jriver(echo), NE2000, downstream, tcp

1: elephant(1047) -> jriver(echo), NE2000, downstream, tcp

Filtered Packets 14:36:00

1: elephant(1047) -> jriver(echo), NE2000, downstream, tcp

These show that a fragmentation overwrite attack was filtered at 2:35:45 P.M., coming from a probably spoofed IP address on the SMTP port. (Remember that these attacks are not filtered by normal rules). Also, several echo packets were filtered at 2:35 and 2:36 from the host elephant.

## E-mail Alerts

The *icewatch* program enables ICE.Block to send an E-mail alert to one or more E-mail addresses when certain data packet filtering events are detected. Multiple instances of *icewatch* can be executed, each monitoring a different event.

The *icewatch* program can be added to the */etc/iceblock/iceblock.rc* script to run when the system boots, or run manually.

A filtering event occurs when a specified number of packets (*threshold*) are filtered within a specified interval of time. (The default is 300 seconds, or five minutes.) For example, a threshold of 1 sends an E-mail each time a packet is filtered.

The *icewatch* program also can be configured to look for packet filtering events from specific sources, destinations, and destination ports, and to send E-mails to several addresses.

E-mail alerts are easily defined using the Windows 95 GUI program or command line interface.



## Configuring E-mail Alerts Using the GUI Program

To set up E-mail alerts using the Windows 95 GUI program, click the Logging button to display the Logging window. In the top half of the Logging window is a sub-window controlled by three tabs that let you toggle between dialog boxes in which you can set up reports (as described above), set up E-mail alerts, and activate console logging.

Each E-mail alert you set up is displayed in the lower half of the Logging window.

To set up a new E-mail alert, click on the tab labeled Alert. In the dialog box, enter the parameters that determine when an E-mail alert will be sent, including the number of violations required (*threshold*) and a *time interval*; a *source*, *destination*, and *port* to monitor for violations; and an *E-mail address* where alerts should be sent.

GETTING  
STARTED

Figure 3.14: Logging Window with Alert Tab Selected

Type	From	To	Port	Sort	Thresh	E-mail
Report	*	secret.jriver.com	80	Source	5	report@jriver.com
Report	ftp.cdrom.com	ftp.jriver.com	21	Source	5	report@jriver.com
Alert	*	backups.jriver.com	*	N/A	5	alert@jriver.com
Alert	intern.jriver.com	www.theonion.com	80	N/A	5	alert@jriver.com

Configure Logging



Figure 3.15: Dialog Box Displayed Using Alert Tab

After entering the alert parameters, click the Add button to add the new alert entry to the list.

To change the parameters of an existing alert entry, click in the listing to highlight it, click the Edit button to display it in the dialog box, make the changes, and click Submit.

To delete an existing alert entry, click in the listing to highlight it, then click the Remove button.

**Console Logging.** When *console logging* is activated, warning messages are displayed immediately on the firewall console. To activate (or deactivate) console logging, click on the tab labeled Console Logging, then click on the appropriate radiobutton to Enable or Disable it.

Refer to the next section for more detailed information on alert parameters, as defined using command line editing.

Figure 3.16: Dialog Box Displayed Using Console Tab



## Configuring E-mail Alerts Using Command Line Editing

At the command line, settings are changed by adding command options to the `icewatch` command line. The syntax is:

```
icewatch -ttime -ssource -ddest -pport -Ddir -llag threshold user(s)
```

### **-ttime**

This command option specifies the *time interval* to check. (The default is 300 seconds.) When the number of packets filtered during the specified time interval exceeds the threshold, the E-mail is sent.

### **-ssource**

This command option defines a specific *source address*, as defined in the `iceblock.conf` configuration file. The default is all sources.

### **-ddest**

This command option defines a specific *destination address*, as defined in the `iceblock.conf` configuration file. The default is all destinations.

### **-pport**

This command option defines a specific *destination port*, as defined in the `iceblock.conf` configuration file.

### **-Ddir**

This command option specifies the *directory* where the log files are stored. (The default is `/etc/iceblock/log`.) This should be the same directory used by `iceblockd` (the logging daemon) and `icelog` (the reporting program). If you change the directory where `iceblockd` places log files, make the same change here.

### **-llag**

This command option defines *lag time*, which should correspond with the time interval setting used by the logging daemon `iceblockd`. (The default is 15 seconds.) For example, if this value is changed for `iceblockd`, it must be changed here as well. The interval defined here must always be the same or



larger than the value used by iceblockd. If the lag time is shorter, icewatch will search for filtering events before iceblockd has created the data, and thus assume that no events have occurred.

### ***threshold***

This refers to the number of violations required to trigger an alert. It works in conjunction with the specified time interval: for example, if a threshold setting of 10 is used in conjunction with the default time interval of 300 seconds, an E-mail alert would be sent when 10 of the specified data packets are filtered within a period of five minutes. A threshold setting of 1 sends an E-mail each time a packet is filtered

### ***user(s)***

This is a list of addresses for E-mail recipients.

## **Example**

To determine if any 30 packets are filtered in one minute (a sign of a possible *door twisting attack*), or if five packets destined for telnet are filtered in 10 seconds (a possible *password cracking attack*), or if any packet from host badguys.foo.com (a known evildoer site) is detected (checking every five seconds), we would use the following:

```
icewatch -t 60 30 root@myhost
icewatch -t 10 -p telnet 5 root@myhost
icewatch -t 5 -s badguys.foo.com 1 root@myhost
```

