

Tivoli Enterprise



Firewall Security Toolbox User's Guide

Version 1.3.2

Tivoli Enterprise



Firewall Security Toolbox User's Guide

Version 1.3.2

Note

Before using this information and the product it supports, read the information in "Notices" on page 85.

July 2008

This edition applies to version 1, release 3, modification 2 of Tivoli Enterprise Firewall Security Toolbox (product number 5698-FRA) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2003, 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	v
Who should read this guide	v
What this guide contains	v
Prerequisite and related publications	v
Tivoli Management Framework library	vi
Related publications	vi
Accessing publications online	vii
Ordering publications	vii
Accessibility	vii
Contacting software support	vii
Conventions used in this guide	viii

Chapter 1. Introduction	1
Tivoli environment with a firewall	1
Tivoli environment with demilitarized zones.	2
Firewall limitations on connectivity	3
Sending events across firewalls	4
Identifying a component as parent or child	5

Chapter 2. Installing Firewall Security	
Toolbox	7
Prerequisite software	7
Planning where to install components	8
Getting started.	8
Components on multihomed hosts	9
Decompressing the installation files	10
Installing on UNIX operating systems	10
Installing a UNIX endpoint proxy	10
Installing a UNIX gateway proxy	11
Installing a UNIX relay	12
Installing a UNIX event sink.	14
Installing on Windows operating systems	15
Installing a Windows endpoint proxy.	15
Installing a Windows gateway proxy	17
Installing a Windows relay	20
Installing additional relays on the same machine	23
Installing a Windows event sink	24
Upgrading the components	25
Upgrading on Windows operating systems.	25
Upgrading on UNIX operating systems	26
Uninstalling the components	26
Uninstalling from UNIX operating systems	26
Uninstalling all components except the relay from Windows operating systems	26
Uninstalling the relay from Windows operating systems.	27

Chapter 3. Configuring the components 29	
Configuring the endpoint proxy	29
[endpoint-proxy] section	29
[log] section	30
[communication-layer] section	31
[children-cm-info] section.	32
Configuring the gateway proxy.	34

[gateway-proxy] section	34
[log] section	35
[communication-layer] section	36
[parent-cm-info]	37
Configuring the relay	38
[relay] section	39
[log] section	39
[communication-layer] section	40
[children-cm-info] section.	41
[parent-cm-info] section	42
Configuring the event sink	43
[SENDING] section.	44
[RECEPTION] section	44
[EIF] section	45
[LOG] section.	45
Configuring non-TME adapters for the event sink	46
Migrating endpoints to connect to a gateway proxy	46
Configuring backup gateway proxies	47
Configuring endpoints for backup gateway proxies	48
Configuring enable-identity	49

Chapter 4. Using Firewall Security	
Toolbox	53
Starting and stopping the components	53
Starting and stopping the components on Windows operating systems	53
Starting and stopping the components on UNIX operating systems	53
Working with endpoints logged in through the proxy	54
Listing the endpoints in the database.	54
Modifying the attributes of an endpoint	55
Reassigning an endpoint to a new gateway proxy	55
Changing the port on the endpoint	55
Reassigning an endpoint to a new endpoint proxy	56
Removing an endpoint from the database	56
Backing up and restoring the endpoint manager database	57
Installing endpoints in a DMZ	57
Installing the endpoints from scratch	57
Connecting endpoints that are already present in the Tivoli region.	57
Processing events from the Tivoli Enterprise Console Availability Intermediate Manager console	58
Viewing endpoint properties.	59

Appendix A. Using the command line interface	61
Command line syntax	61
wproxy.	62

Appendix B. Troubleshooting	67
Testing proxy configuration	67

Debugging application errors	68
Using log files for troubleshooting.	68
Providing more detail in the log files.	68
Interpreting the log files	70
Providing details to customer support	71
Tuning	72
Tuning for large distributions	72
Tuning to reduce distribution timeouts	72
Timeout values for Tivoli Management Framework	72
Timeout Values for the Firewall Security Toolbox.	73
Connecting components from different versions	73
Rescuing lost endpoints from the gateway	73
Error when installing as user nobody.	73
Thread shortage on UNIX operating systems	74
Network Address Translation support	75
Wake on LAN not supported	75
Gateway proxy label might be displayed incorrectly	75
Multicast feature not supported	75
Port conflicts	75

Gateway times out before distribution completed.	75
--	----

Appendix C. Configuring ports for Firewall Security Toolbox. 77

Ports used by Firewall Security Toolbox	77
The local-port-range and port-range parameters	77
Configuring source ports	78
Configuring firewalls	80

Appendix D. Understanding communication packets 81

Understanding communication protocol	81
Inspecting Tivoli Management Framework packets	82

Notices 85

Trademarks	86
----------------------	----

Index 87

Preface

Tivoli Enterprise Firewall Security Toolbox, referred to as the Firewall Security Toolbox in the rest of this book, provides a solution for managing your Tivoli® network across firewalls without compromising security. This guide explains how to install and configure this feature of Tivoli Management Framework.

Who should read this guide

This guide is for administrators and system programmers who configure the firewalls in their networks, but it is also useful for network planners who organize the security configuration of their networks. Readers should be familiar with the following:

- The UNIX® and Windows® operating systems
- Tivoli Management Framework

What this guide contains

This guide contains the following sections:

- Chapter 1, “Introduction,” on page 1
Provides an overview of the main concepts.
- Chapter 2, “Installing Firewall Security Toolbox,” on page 7
Provides instructions for installing the components.
- Chapter 3, “Configuring the components,” on page 29
Explains how to configure the components.
- Chapter 4, “Using Firewall Security Toolbox,” on page 53
Explains how to perform various tasks, including starting and stopping the components.
- Appendix A, “Using the command line interface,” on page 61
Provides detailed usage information for the **wproxy** command.
- Appendix B, “Troubleshooting,” on page 67
Provides information to help identify and solve problems, including how to interpret the log files.
- Appendix C, “Configuring ports for Firewall Security Toolbox,” on page 77
Explains how Firewall Security Toolbox uses ports, provides information to help you configure these ports, and describes how to configure firewalls between Firewall Security Toolbox components.
- Appendix D, “Understanding communication packets,” on page 81
Provides information about the communication packets used by Firewall Security Toolbox.

Prerequisite and related publications

This section lists publications in the Tivoli Management Framework library and any other related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to make comments on Tivoli publications.

Tivoli Management Framework library

The following documents are available in the Tivoli Management Framework library:

- *Tivoli Management Framework Planning for Deployment Guide*
Explains how to plan for deploying your Tivoli environment. It also describes Tivoli Management Framework and its services.
- *Tivoli Enterprise Installation Guide*
Explains how to install and upgrade Tivoli Enterprise software within your Tivoli region using the available installation mechanisms provided by Tivoli Software Installation Service and Tivoli Management Framework. Tivoli Enterprise software includes the Tivoli server, managed nodes, gateways, endpoints, and RDBMS Interface Module (RIM) objects. This guide also provides information about troubleshooting installation problems.
- *Tivoli Management Framework User's Guide*
Describes the concepts and procedures for using Tivoli Management Framework services. It provides instructions for performing tasks from the Tivoli desktop and from the command line.
- *Tivoli Management Framework Maintenance and Troubleshooting Guide*
Explains how to maintain a Tivoli environment and troubleshoot problems that can arise during normal operations.
- *Tivoli Management Framework Maintenance and Troubleshooting Guide*
Explains how to maintain a Tivoli environment and troubleshoot problems that can arise during normal operations.
- *Tivoli Management Framework Release Notes*
Describes the latest installation information, including supported platforms, defects, and limitations.
- *Tivoli Management Framework Reference Manual*
Provides in-depth information about Tivoli Management Framework commands. This manual is helpful when writing scripts that are later run as Tivoli tasks. This manual also documents default and validation policy scripts used by Tivoli Management Framework.

Related publications

The following document also provides useful information related to Firewall Security Toolbox:

- *IBM Tivoli Enterprise Console Adapters Guide, GC32-0668*
Provides detailed descriptions for the currently available Tivoli Enterprise Console[®] adapters.
- *IBM Tivoli Remote Control User's Guide, GC31-8437*
Provides information about using the Firewall Security Toolbox with Remote Control.

The *Tivoli Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Glossary* is available, in English only, at the following Web site:

<http://publib.boulder.ibm.com/tividd/glossary/termsmst04.htm>

Accessing publications online

The documentation CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:

<http://publib.boulder.ibm.com/tividd/td/tdprodlist.html>

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File → Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at the following Web site:

<http://www.elink.ibm.link.ibm.com>

From this Web page, select **Publications** and follow the instructions.

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, see the following Web site for a list of telephone numbers:

<http://www.ibm.com/software/tivoli/order-lit>

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

Contacting software support

If you have a problem with any Tivoli product, refer to the following IBM Software Support Web site:

<http://www.ibm.com/software/sysmgmt/products/support/>

If you want to contact software support, see the *IBM Software Support Guide* at the following Web site:

<http://techsupport.services.ibm.com/guides/handbook.html>

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility

- Telephone numbers, depending on the country in which you are located
- Information you must have before contacting IBM Software Support

Conventions used in this guide

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls
- Keywords and parameters in text

Italic

- Words defined in text
- Emphasis of words (words as words)
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples and code examples
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

This guide uses the UNIX convention for specifying environment variables and for directory notation:

- When using the Windows command line, replace *\$variable* with *%variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths.
- When using the bash shell on Windows operating systems, use the UNIX conventions.

Chapter 1. Introduction

A simple Tivoli environment consists of the Tivoli server, a gateway, and endpoints. The endpoints communicate with the Tivoli server through the gateway and the gateway communicates with the Tivoli server. See Figure 1.

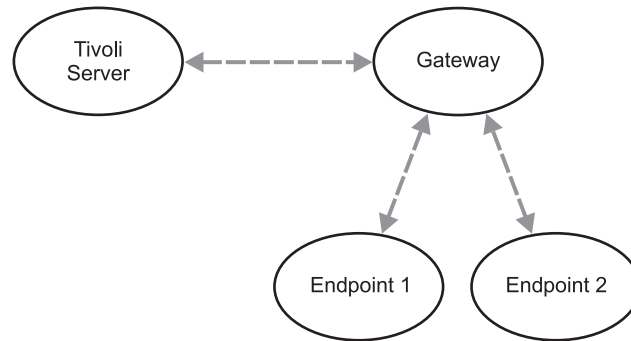


Figure 1. A simple Tivoli environment

Your Tivoli environment can be as simple or complex as your network demands. You can install multiple gateways in a Tivoli region to manage large numbers of endpoints effectively.

When one or more firewalls exist between an endpoint and a gateway, the communication channels permitted by the firewall are limited. Firewall Security Toolbox enables the endpoint and gateway to communicate across firewalls while respecting firewall restrictions.

Tivoli environment with a firewall

On one side of the firewall, Firewall Security Toolbox provides an *endpoint proxy* that connects to the gateway as if it were the endpoints. On the other side of the firewall, the endpoints are connected to a *gateway proxy*, as if it were the gateway. The gateway proxy and endpoint proxy communicate with each other through the firewall. Figure 2 on page 2 shows a simple configuration with one gateway proxy and one endpoint proxy.

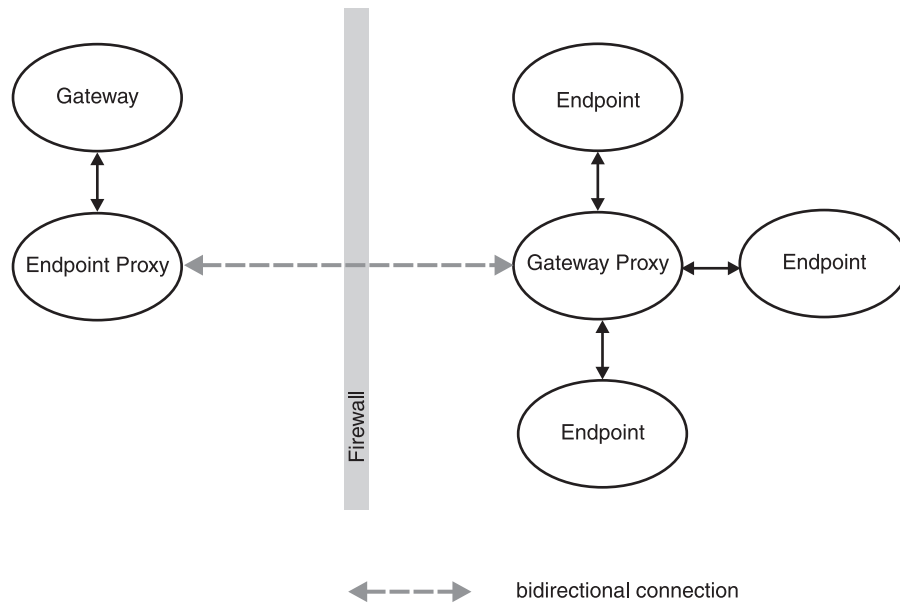


Figure 2. A Tivoli environment with an endpoint proxy and a gateway proxy connecting through a single firewall

Just as multiple endpoints can connect to a single gateway and multiple gateways to a single Tivoli server, multiple endpoints can connect to a single gateway proxy and multiple gateway proxies can connect to a single endpoint proxy. The endpoint proxy emulates all the endpoints to the gateway that manages them.

The communications between these Tivoli components is based on a Tivoli proprietary protocol over TCP/IP.

Tivoli environment with demilitarized zones

When a network includes several firewalls that separate demilitarized zones (DMZs) of progressively lower security as they approach the Internet, the configuration becomes more complex. Although the gateway proxy and endpoint proxy continue to communicate with the endpoint and the gateway, respectively, they no longer communicate directly across the multiple firewalls, because this would defeat the purpose of having multiple firewalls in place.

Instead, Firewall Security Toolbox provides *relays*, which are installed between the firewalls in DMZs. These relays pass on information to each other from one DMZ to another and, finally, to or from the endpoint proxy and gateway proxy. Figure 3 on page 3 shows an example of this configuration.

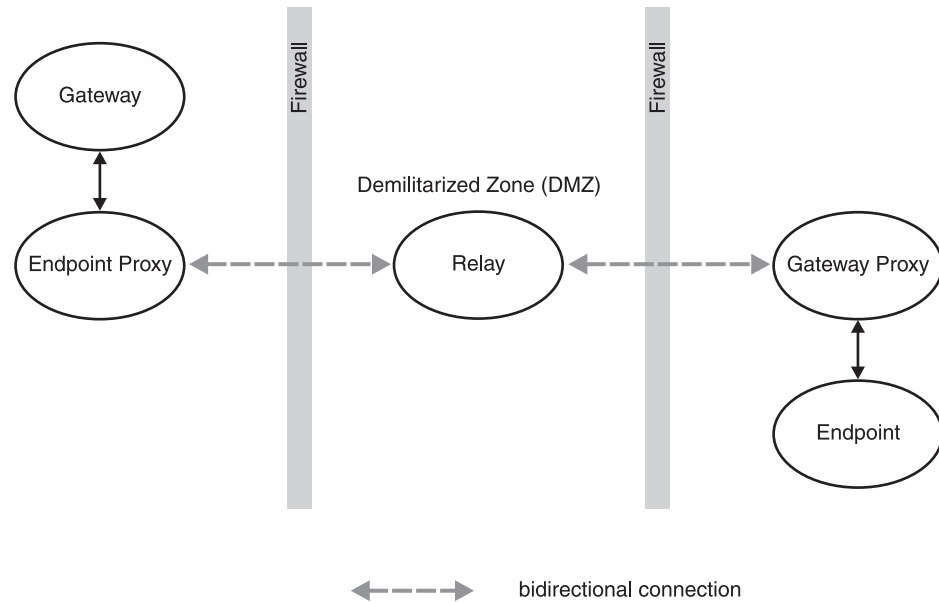


Figure 3. A Tivoli environment with the relay connecting the endpoint and gateway proxies through a DMZ

You can install multiple instances of a relay on a single machine. For all other components of Firewall Security Toolbox, you can install only one instance per machine.

Firewall limitations on connectivity

The firewall can allow connections between machines that have been initiated by either machine. These are known as *bidirectional* connections.

However, this can expose the Tivoli server to illicit connections by unauthorized machines posing as legitimate clients. To avoid such intrusions, each firewall can be configured to limit which machines can initiate a connection. This usually means that the machine on the more secure side initiates all connections with other machines on the less secure side. This machine is known as a client and becomes the *initiator*. The other machine is known as a server and becomes the *listener*. This type of connection is known as *unidirectional*. Firewall Security Toolbox enables you to configure unidirectional connections among the endpoint proxy, gateway proxy, and relays in your Tivoli environment.

In a configuration composed of multiple firewalls and DMZs, you can set up a network that allows a mix of unidirectional and bidirectional connections. For example, a bank branch office, which operates within a secure intranet, connects to its main administrative office through a series of relays (see Figure 4 on page 4).

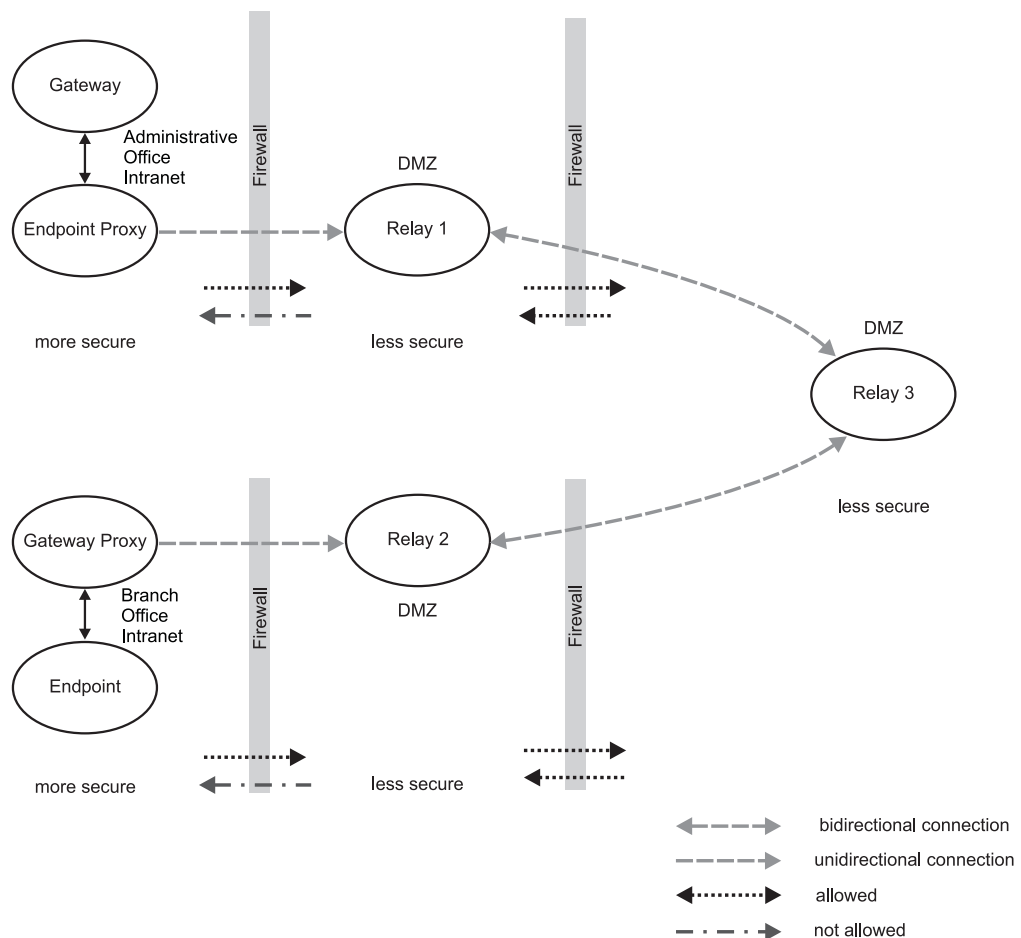


Figure 4. Example of an environment with DMZs, relays, and bidirectional and unidirectional connections

From the more secure sides, the offices use unidirectional connections to the relays 1 and 2. Relays 1 and 2 use bidirectional connections to relay 3 in the less secure areas. In the unidirectional connection between the gateway proxy and relay 2, the gateway proxy is the initiator and relay 2 is the listener.

Sending events across firewalls

TME[®] adapters use endpoints to send events to the Tivoli Enterprise Console server through Tivoli connections. When a firewall separates the endpoint from the Tivoli Enterprise Console server, the machines connect through the gateway and endpoint proxies.

Machines that are not part of the Tivoli environment use non-TME adapters to send events to the Tivoli Enterprise Console server through non-Tivoli connections. When a firewall separates the non-TME adapter machine from the Tivoli Enterprise Console server, Firewall Security Toolbox provides the *event sink*, which sends the events to the Tivoli Enterprise Console gateway. The event sink, which is installed on an endpoint outside the firewall, collects events sent from non-TME adapters as if it were a Tivoli Enterprise Console server and sends them to the Tivoli Enterprise Console server as though they were TME events. The event sink can collect events from multiple non-TME adapters. See Figure 5 on page 5.

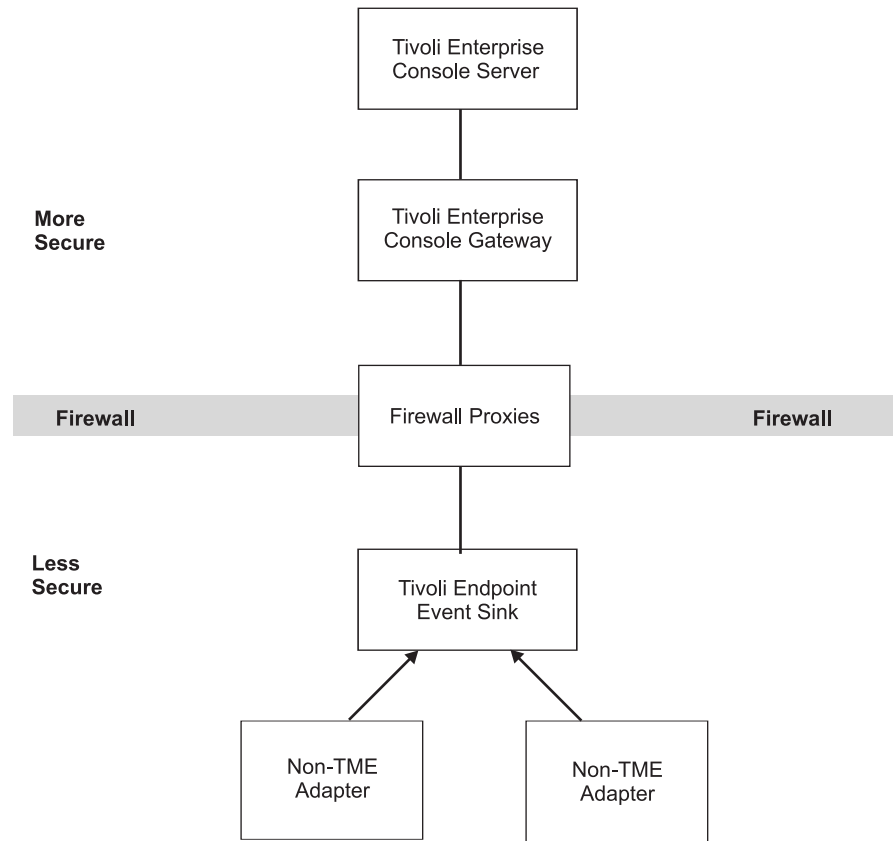


Figure 5. an event sink sends non-TME events to the Tivoli Enterprise Console server through the firewall proxies

The event sink can also collect Tivoli events that are sent from Tivoli Distributed Monitoring or the Tivoli Enterprise Console Availability Intermediate Manager console and forward them to Tivoli Enterprise Console servers across firewalls.

Identifying a component as parent or child

The hierarchy of the components of Firewall Security Toolbox is presented in terms of a *parent* and *children*. The endpoint proxy, the parent, can have one or more children, either relays or gateway proxies. An endpoint proxy does not have a parent because it connects to the gateway.

A relay can have a parent (an endpoint proxy or another relay) and children (relays or gateway proxies). A gateway proxy has a parent but no children because a gateway proxy connects to endpoints.

For example, in Figure 3 on page 3, the endpoint proxy is the parent and the relay is its child. The relay is the parent of the gateway proxy and the gateway proxy is its child. The endpoint proxy and relays can have more than one child (relay or gateway proxy) but each component has only one parent. Because the gateway proxy is at the bottom of this hierarchy, it has no children. In another example (Figure 4 on page 4), relay 1 is the child of the endpoint proxy and the parent of relay 3. Relay 3 is the parent of relay 2. Relay 2 is the parent of the gateway proxy. Understanding this hierarchy is important when you install and configure the components.

Chapter 2. Installing Firewall Security Toolbox

This chapter explains how to install and configure the components of Firewall Security Toolbox.

Prerequisite software

Firewall Security Toolbox is a feature of Tivoli Management Framework that was introduced in Version 3.7.1.

The event sink must have an endpoint from Tivoli Management Framework, Version 4.3.1 installed, and the event sink must be installed in an environment where IBM® Tivoli Enterprise Console, Version 3.9 is installed.

All other components of Firewall Security Toolbox do not require any Tivoli Enterprise software.

The following table lists the required minimum operating system software to run the toolbox components.

Table 1. Prerequisite Operating System Software

AIX®	AIX Version 5.1, 5.2, 5.3, 6.1 in 32-bit mode
HP-UX	HP-UX Version 11.0, 11v1, 11v2, 11v3 in 32-bit mode
Linux	<ul style="list-style-type: none">• Linux-ix86: Red Hat 3, 4, 5; SLES 8, 9, 10• Linux-ppc: Red Hat 3, 4, 5; SLES 8, 9, 10• Linux-s390: Red Hat 3, 4, 5; SLES 8, 9, 10• Red Hat Linux for Intel, interpreter i386, Version 7.2 (Kernel 2.4.7-10smp), 7.3, 8.0, Advanced Server 2.1, or Advanced Server 3.0• SuSE Linux Version 8.1, SuSE Linux Enterprise Server Version 7, or SuSE Linux Enterprise Server Version 8.• United Linux Version 1.0.• Linux on zSeries
Microsoft® Windows	<ul style="list-style-type: none">• Windows XP Professional• Windows Server 2000, Service Pack 3 or 4• Windows Server 2000 Advanced Server, Service Pack 3 or 4• Windows Server 2000 Datacenter Server, Service Pack 3 or 4• Windows Server 2003, Standard Edition• Windows Server 2003, Enterprise Edition• Windows Server 2003, Datacenter Edition• Windows Vista• Windows Server 2008
Sun Solaris Operating System	<ul style="list-style-type: none">• SPARC: Solaris 8, 9, 10 in 32-bit mode• ix86: Solaris 8, 9, 10

Planning where to install components

You can install as many gateway proxies and endpoint proxies as you need in the firewall region. When considering the number of endpoints per endpoint proxy, you must consider the overall usage and load placed on the systems. An endpoint proxy can only support as many endpoints as are supported by the TMF gateway that the endpoint proxy communicates with. A maximum of 500 endpoints per endpoint proxy is recommended. However, this is not an architectural limit. Your limit might be different depending on the demands of your environment.

An endpoint proxy emulates endpoints to only one gateway. However, you can use multiple endpoint proxies per gateway.

The total number of endpoints that an endpoint proxy can emulate depends on the number of endpoint proxies per gateway and the limitations of the gateway itself. The recommended limit for endpoints communicating with one gateway is 2000. (See the *Tivoli Management Framework Planning for Deployment Guide*.) This limit applies to endpoints emulated by an endpoint proxy. For example, for a gateway that supports 2000 endpoints, you could use eight endpoint proxies that each support 250 endpoints, or four endpoint proxies that each support 500 endpoints.

Install multiple gateway proxies in a DMZ to provide backup gateway proxies when the main gateway proxy is unavailable. See “Configuring backup gateway proxies” on page 47 for configuration details.

You can install multiple instances of the relay component on a single machine. However, you can install only one instance of the endpoint proxy, gateway proxy, and event sink per machine.

You cannot run different components on the same machine and use them to connect to the same component on another machine. For example, you cannot have a relay and a gateway proxy on a machine and use them to connect to the same endpoint proxy. You can connect the gateway proxy to the relay on the same machine and the relay to an endpoint proxy on another machine.

Install a few endpoints first to test connectivity from the Tivoli region to an endpoint through the proxy. Enter the following command from the Tivoli server or managed node:

```
wadminep endpoint view_config_info
```

where *endpoint* is the label of the endpoint.

If you cannot reach the endpoint, follow the instructions in “Testing proxy configuration” on page 67.

Because the toolbox requires a particular configuration of the Tivoli region, keep machines that are in less secure zones in separate Tivoli regions. Set up a separate Tivoli region to manage resources that are in a DMZ. To manage your endpoints as if they were in a single Tivoli region, you can interconnect the firewall region to non-firewall regions.

Getting started

You need to do the following to get started:

- Ensure that the components of Firewall Security Toolbox that will communicate with each other directly have IP visibility of each other. Depending on your configuration, these components can be the endpoint proxy and gateway proxy, a proxy and a relay, or two relays. You can use DNS if you have DNS configured. However, there is no requirement to use DNS host names. The TCP/IP address works as well. TCP/IP connectivity is required. If you use the DNS name of the machine, ensure that the DNS name of the destination machine is resolved into its IP address.
- Before installing the software, ensure that you have the following information:
 - The port number of the gateway that the endpoint proxy will use to communicate.
 - The host name or IP address of all the components that you are installing.
- Decide on some additional ports that the components will use to communicate with each other:
 - The endpoint proxy requires a range of ports used to emulate the endpoints logged in through Firewall Security Toolbox.
 - Gateway proxies require one port to receive traffic from the endpoint proxy or relay and another port to listen for traffic from the endpoints.
 - Relays require ports to receive traffic from the components with which they connect, one for the parent and one for the children.

Use the following criteria to choose the port number:

- The port must not be used by other applications
- The user account that you specify must have the privileges to use the port

For more information about Firewall Security Toolbox and ports, see Appendix C, “Configuring ports for Firewall Security Toolbox,” on page 77.

Components on multihomed hosts

When a machine has more than one network interface and address, it is known as a *multihomed* host. Multihomed hosts might need to connect to one component in one subnet and another component in another subnet. For example, an endpoint proxy machine might connect to a gateway in one subnet and relays or gateway proxies in another subnet (see Figure 6 on page 10).

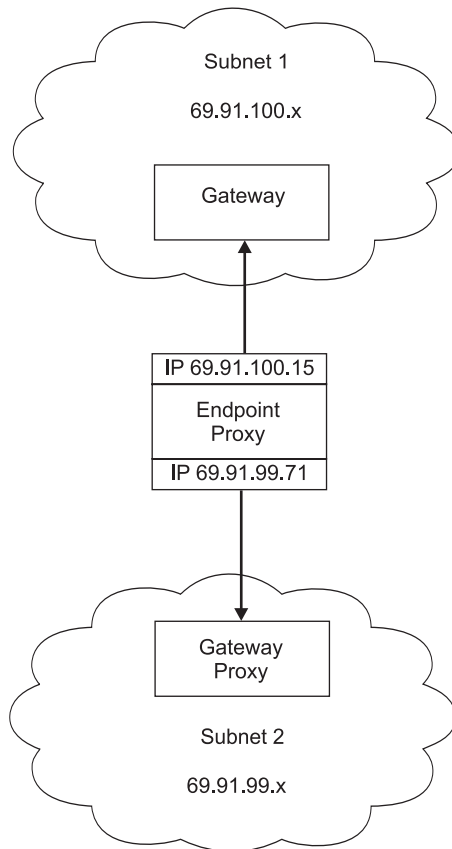


Figure 6. Example of a multihomed host

The endpoint proxy can connect to the gateway using the DNS name or IP address of one network interface, for example, 69.91.100.15, and to the relays or gateway proxies using the DNS name or IP address of another network interface, for example, 69.91.99.71. When configuring the components with multihomed hosts, you need to specify the correct DNS name or IP address. See Chapter 3, “Configuring the components,” on page 29 for more details.

Decompressing the installation files

To install Tivoli Enterprise Firewall Security Toolbox, decompress the 1.3.2-TFS-0001.tar file. Under the main \Proxy directory, the file creates directories for each component and copies installation scripts to subdirectories for each platform.

Installing on UNIX operating systems

The following sections describe how to install the components on UNIX operating systems. These operations need to be run as root user.

Installing a UNIX endpoint proxy

To install the endpoint proxy, follow these steps:

1. From the \EndpointProxy directory, go to the directory for the platform on which the proxy will run.
2. Run the ./install.sh script.
3. Provide the following information:

- a. To install the Tivoli Endpoint Proxy, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:
Enter Y to accept the license agreement and continue the installation.
- b. Installation directory [default=/usr/epp]:
Specify the directory where you want to install the endpoint proxy. If the directory does not exist, you are prompted to create it.
- c. Run the endpoint proxy as the following user:
Specify the account name to use to run the proxy process. This account must exist and it is recommended that you use an unprivileged account.
- d. Gateway hostname:
Specify the host name or IP address of the gateway with which the endpoint proxy communicates. The endpoint proxy can communicate with only one gateway in a Tivoli region.
- e. Gateway port [default=9494]:
Specify the TCP/IP port number of the gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. This is normally port 9494. Do not change this value unless the gateway is known to be using a different listening port with the endpoint.
- f. Endpoint Proxy Port:
Specify the port number of the endpoint proxy machine from which it listens for connections with the relay or gateway proxy.
- g. Relay or gateway proxy hostname:
Specify the host name of the relay or gateway proxy with which the endpoint connects.
- h. Relay or gateway proxy port:
Specify the port number from which the relay or gateway proxy listens for connections from the endpoint proxy.
- i. Enter more destinations? [Y/N] [default=N]:
Enter Y to specify additional relays or gateway proxies with which the endpoint connects. You are asked to specify the host name and port number of the additional destination machines. When you are done adding destination machines and are ready to continue with installation, enter N.
- j. Specify how endpoint proxy connects to destinations [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:
Enter 0 to permit connections that are initiated by either machine.
Enter 1 to permit connections initiated by the endpoint proxy only.
Enter 2 to permit connections initiated by the destination machine only.

Note: The mode you select here applies to all of the children for the endpoint proxy. The endpoint proxy must connect to all of its children using the same connection mode.
- k. Start the endpoint proxy? [Y/N] [default=Y]:
Enter Y to start the endpoint proxy. Otherwise, enter N. To start the component at another time, see “Starting and stopping the components” on page 53.

Installing a UNIX gateway proxy

To install the gateway proxy, follow these steps:

1. From the \GatewayProxy directory, go to the directory for the platform on which the proxy will run.
2. Run the ./install.sh script.
3. Provide the following information:
 - a. To install the Tivoli Gateway Proxy, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:
Enter Y to accept the license agreement and continue the installation.
 - b. Installation directory [default=/usr/gwp]:
Specify the directory where you want to install the gateway proxy. If the directory does not exist, you are prompted to create it.
 - c. Run the gateway proxy as the following user:
Specify the account name to use to run the proxy process. This account must exist and it is recommended that you use an unprivileged account.
 - d. Name (label) for this proxy [default=localhost]:
Optionally, enter a name to identify the gateway proxy.
 - e. Port to listen on for TMA traffic [default=9494]:
Enter the port number on the gateway proxy that represents the gateway to the endpoints. The default is 9494.
 - f. Gateway proxy port:
Specify the port number that the gateway proxy uses to listen for connections from the relay or endpoint proxy.
 - g. Relay or endpoint proxy hostname:
Specify the host name of the machine that the gateway proxy will connect up the chain toward the Tivoli gateway.
 - h. Relay or endpoint proxy port:
Enter the port number of the machine that the gateway proxy will connect up the chain toward the Tivoli gateway.
 - i. Specify how gateway proxy connects to destination [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:
Specify how the gateway proxy connects to the destination relay or endpoint proxy.
Enter 0 to permit connections that are initiated by either machine.
Enter 1 to permit connections initiated by the gateway proxy only.
Enter 2 to permit connections initiated by the parent machine only.
 - j. Start the gateway proxy? [Y/N] [default=Y]:
Enter Y to start the gateway proxy. Otherwise, enter N. To start the component at another time, see "Starting and stopping the components" on page 53.

Installing a UNIX relay

To install the relay, follow these steps:

1. From the \Relay directory, go to the directory for the platform on which the relay will run.
2. Run the ./install.sh script.
3. Provide the following information:
 - a. To install the Tivoli Relay, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:

Enter Y to accept the license agreement and continue the installation.

- b. Installation directory [default=/usr/relay-1]:

Specify the directory where you want to install the relay. If the directory does not exist, you are prompted to create it.

Note: The default directory ends with a directory named for the numbered instance of the relay. You must launch all operations (start, stop, and uninstall) on an instance of the relay from the directory in which you install the instance.

If you change the default directory, ensure that it is different from the directory where you have installed any previous instances of the relay.

- c. Run the relay as the following user:

Specify a user name to use to run the relay. This account must exist and it is recommended that you use an unprivileged account.

- d. *Relay-Parent Connection Options*

Relay port:

Enter the port number for the relay to communicate with the parent machine.

- e. Relay or endpoint proxy hostname:

Enter the host name for the parent relay or endpoint proxy with which the relay will communicate.

- f. Parent Remote Port:

Enter the port number for the parent relay or endpoint proxy with which the relay will communicate.

- g. Specify how relay connects to parent relay or endpoint proxy [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:

Specify how the relay connects to the destination relay or endpoint proxy.

Enter 0 to permit connections that are initiated by either machine.

Enter 1 to permit connections initiated by the relay only.

Enter 2 to permit connections initiated by the parent machine only.

- h. *Relay-Children Connection Options*

Relay port:

Enter the port number for the relay to communicate with the children machines.

- i. Relay or gateway proxy hostname:

Specify the host name of the child machine, relay or gateway proxy, with which the relay connects.

- j. Relay or gateway proxy port:

Enter the port number of the machine with which the relay connects.

- k. Enter more destinations? [Y/N] [default=N]:

Enter Y to specify additional relays or gateway proxies with which the relay connects. You are asked to specify the host name and port number of the additional destination machines. When you are done adding destination machines and are ready to continue with installation, enter N.

- l. Specify how the relay connects to the child destination
[0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:
Specify how the relay connects to the child relay or gateway proxy.
Enter 0 to permit connections that are initiated by either machine.
Enter 1 to permit connections initiated by the relay only.
Enter 2 to permit connections initiated by the destination machine only.

Note: The mode you select here applies to all of the children for the relay.
The relay must connect to all of its children using the same connection mode.
- m. Start the relay? [Y/N] [default=Y]:
Enter Y to start the relay. Otherwise, enter N. To start the component at another time, see “Starting and stopping the components” on page 53.

To install an additional relay on the same machine, repeat this procedure.

Installing a UNIX event sink

To install the event sink, perform the steps, below. The event sink must be installed on a 4.3.1 endpoint:

1. From the \EventSink directory, go to the directory for the platform on which the proxy will run.
2. Run the ./install.sh script.
3. Provide the following information:
 - a. To install the Tivoli Event Sink, you must accept the agreement written in the License file. If you accept the agreement, enter Y.
[Y/N]:
Enter Y to accept the license agreement and continue the installation.
 - b. Installation directory [default=/usr/eventsink]:
Specify the directory where you want to install the event sink. If the directory does not exist, you are prompted to create it.
 - c. LCF_DATDIR directory:
Specify the LCF_DATDIR directory of the endpoint on which you are installing the event sink.
 - d. Run the event sink as the following user:
Specify a user name to use to run the event sink. This account must exist and it is recommended that you use an unprivileged account.
 - e. Listening Port [default=9444]:
Enter the port number on the endpoint where the event sink will receive events.
 - f. Maximum Number of Events in Package [default=50]:
Enter the maximum number of events that the event sink will send to the Tivoli Enterprise Console server in a single package.
 - g. Maximum Buffer Size [default=40000]:
Enter the maximum buffer size, in bytes, of the package that the event sink will send to the Tivoli Enterprise Console server.
 - h. Start the event sink? [Y/N] [default=Y]:
Enter Y to start the event sink. Otherwise, enter N. To start the component at another time, see “Starting and stopping the components” on page 53.

Installing on Windows operating systems

Firewall Security Toolbox provides a self-extracting EXE file to install each component on Windows operating systems. The installation files are unpacked into a default directory, which you can change. You need to specify this directory only the first time you run this file. You can use these files for any future installations.

You can install in one of the following ways:

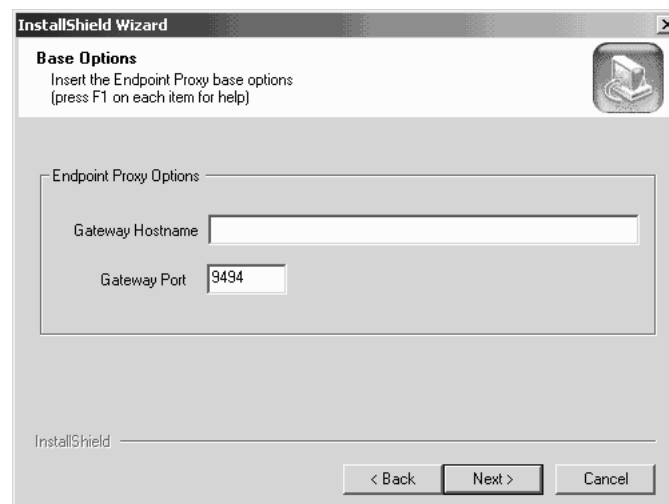
- By running the self-extracting EXE file
- By running setup.exe if you are using a disk image

You must be a member of the Windows Administrators group to perform an installation on a Windows system.

Installing a Windows endpoint proxy

To install the endpoint proxy, do the following:

1. From the directory that contains the Tivoli Endpoint Proxy\w32-ix86\ subdirectory, double-click the Tivoli Endpoint Proxy.exe file. The Tivoli Endpoint Proxy InstallShield Wizard starts.
2. Click **Next**.
3. On the next dialog, enter the directory where the installation files are to be saved and then click **Next**.
4. On the next dialog, click **Next**.
5. On the next dialog, click **Yes** to accept the license agreement.
6. On the next dialog, perform one of the following steps:
 - If a TFSRsrvd account already exists on the machine on which you are installing the endpoint proxy, you are prompted to enter the password. Enter the password and then click **Next**.
 - If a TFSRsrvd account does not exist, you are prompted to create an account:
 - a. Enter a password for this account in the **Password** field.
 - b. Enter the password again in the **Verify** field.
 - c. Click **Next**.
7. On the next dialog, enter the installation directory and then click **Next**. The dialog for Endpoint Proxy Options is displayed.



8. Complete the following fields:

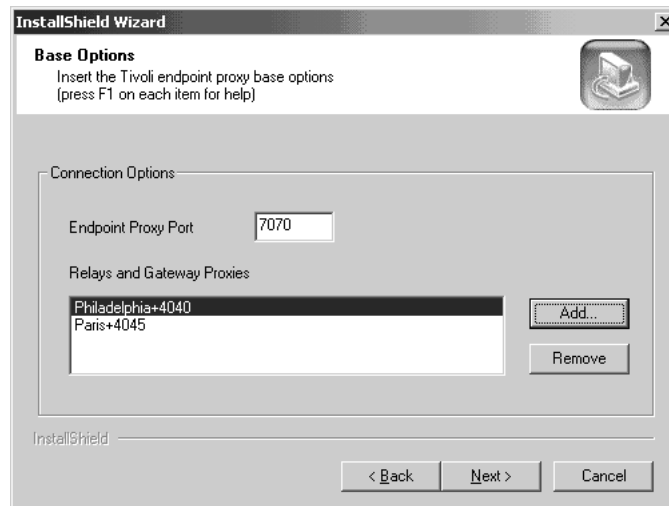
Gateway Hostname

Enter the host name or IP address of the gateway with which the endpoint proxy communicates. The endpoint proxy can communicate with only one gateway in a Tivoli region.

Gateway Port

Enter the TCP/IP port number of the gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. The default is 9494 and should not be changed unless the gateway is known to be using a different listening port with the endpoint.

Click **Next**. The dialog for Connection Options is displayed.



9. Complete the following fields:

Endpoint Proxy Port

Enter the port number of the endpoint proxy machine from which it listens for connections with the relay or gateway proxy.

Relays and Gateway Proxies

Lists the relays and gateway proxies with which the endpoint proxy connects.

To add a relay or gateway proxy to the list of destination machines, click **Add**. The Add Relay or Gateway Proxy dialog is displayed.



Complete the fields and click **OK**:

Hostname

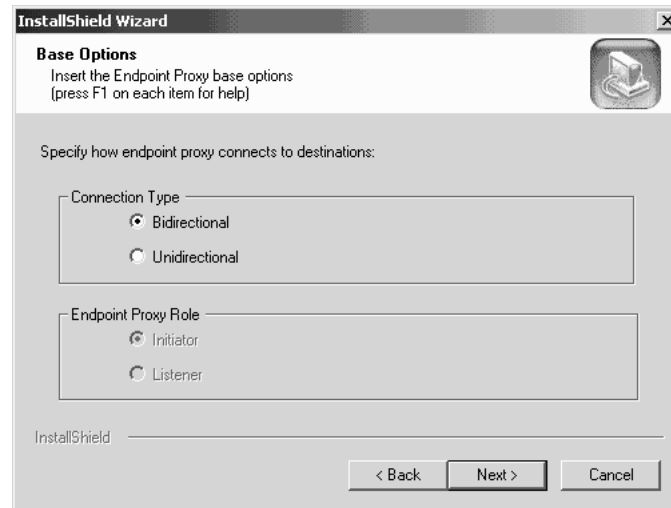
Enter the host name of the relay or gateway proxy with which the endpoint connects.

Port

Enter the port number from which the relay or gateway proxy listens for connections from the endpoint proxy.

To remove a machine, select it and click **Remove**.

Click **Next**. The dialog for the type of endpoint proxy connection is displayed.



10. Specify how the endpoint proxy connects to its children (relays and gateway proxies):

- Select **Bidirectional** to permit connections that are initiated by either machine.
- Select **Unidirectional** to permit connections initiated by only one machine. If you select this option, the **Endpoint Proxy Role** box is enabled:

Initiator

Select to specify that the endpoint proxy machine can start the connection with the destination machines.

Listener

Select to specify that the destination machines can start the connection with the endpoint proxy machine.

11. Click **Next**. The dialog shows a summary of your input.
12. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the endpoint proxy.
13. A message asks you whether or not you want to start the endpoint proxy. Click **Yes** to start it. Otherwise, click **No**. To start the component at another time, see “Starting and stopping the components” on page 53.

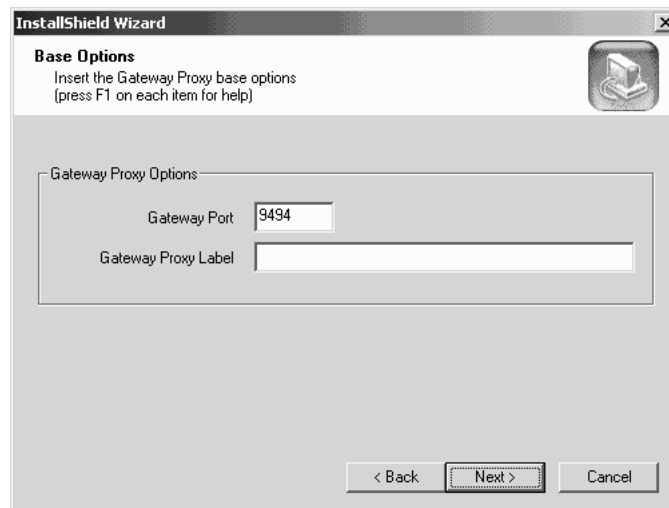
Installing a Windows gateway proxy

The gateway proxy needs to be installed on a host that is in the DMZ where the endpoints will be located.

To install the gateway proxy, do the following:

1. From the directory that contains the gateway Proxy\w32-ix86\ subdirectory, double-click the Tivoli Gateway Proxy.exe file. The Tivoli Gateway Proxy InstallShield Wizard starts. Click **Next**.
2. Click **Next**.
3. On the next dialog, enter the directory where the installation files are to be saved and then click **Next**.
4. On the next dialog, click **Next**.

5. On the next dialog, click **Yes** to accept the license agreement.
6. On the next dialog, perform one of the following steps:
 - If a TFSRsrvd account already exists on the machine on which you are installing the gateway proxy, you are prompted to enter the password. Enter the password and then click **Next**.
 - If a TFSRsrvd account does not exist, you are prompted to create an account:
 - a. Enter a password for this account in the **Password** field.
 - b. Enter the password again in the **Verify** field.
 - c. Click **Next**.
7. On the next dialog, enter the installation directory and click **Next**. The dialog for Gateway Proxy Options is displayed.



8. Complete the following fields:

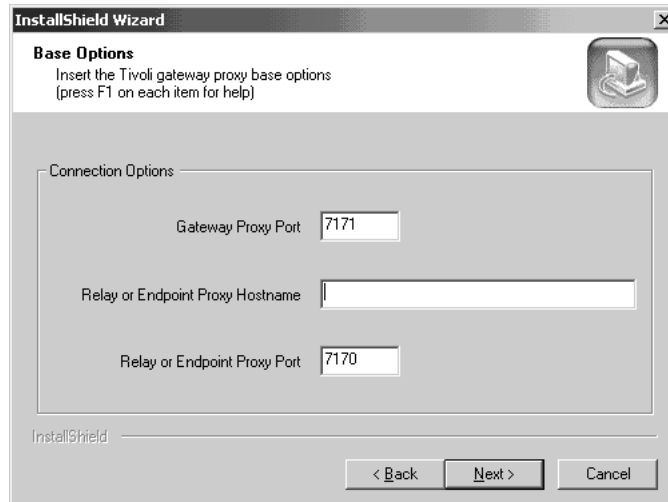
Gateway Port

Enter the port number on the gateway proxy that represents the gateway to the endpoints. The default is 9494.

Gateway Proxy Label

Optionally, enter a name to identify the gateway proxy.

Click **Next**. The dialog for Gateway Proxy-Parent Connection Options is displayed.



9. Complete the following fields:

Gateway Proxy Port

Enter the port number that the gateway proxy uses to listen for connections from the relay or endpoint proxy.

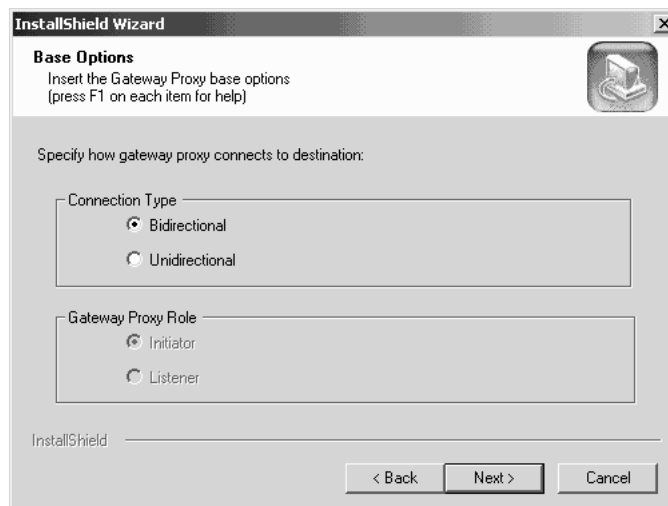
Relay or Endpoint Proxy Hostname

Enter the host name of the machine that the gateway proxy will connect to up the chain toward the gateway.

Relay or Endpoint Proxy Port

Enter the port number of the machine that the gateway proxy will connect to up the chain toward the gateway.

Click **Next**. The dialog for the type of gateway proxy connection is displayed.



10. Specify how the gateway proxy connects to the destination relay or endpoint proxy:

- Select **Bidirectional** to permit connections that are initiated by either machine.
- Select **Unidirectional** to permit connections initiated by only one machine. If you select this option, the **Gateway Proxy Role** box is enabled:

Initiator

Select to specify that the gateway proxy machine can start the connection with the parent endpoint proxy or relay machine.

Listener

Select to specify that the parent machine can start the connection with the gateway proxy machine.

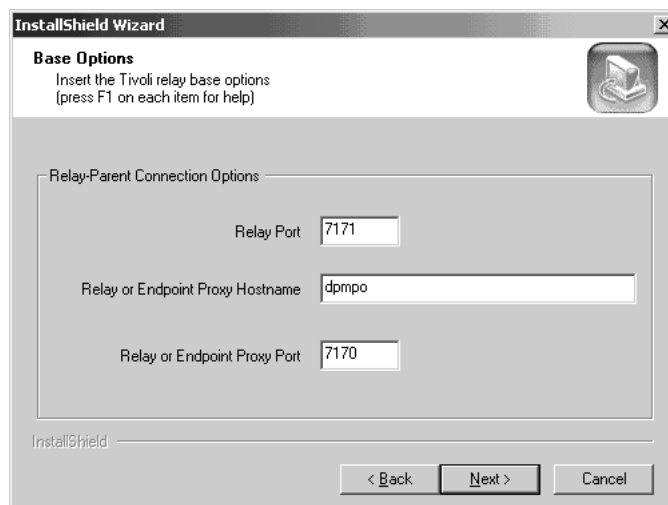
Click **Next**. The dialog shows a summary of your input.

11. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the gateway proxy.
12. A message asks you whether or not you want to start the gateway proxy. Click **Yes** to start it. Otherwise, click **No**. To start the component at another time, see “Starting and stopping the components” on page 53.

Installing a Windows relay

You can install multiple instances of a relay on a single machine. To install the first relay, do the following:

1. From the directory that contains the Tivoli Relay installation images, double-click the Tivoli Relay.exe file. The Tivoli Relay InstallShield Wizard starts. Click **Next**.
2. Click **Next**.
3. On the next dialog, enter the directory where the installation files are to be saved and then click **Next**.
4. On the next dialog, click **Next**.
5. On the next dialog, click **Yes** to accept the license agreement.
6. On the next dialog, perform one of the following steps:
 - If a TFSRsrvd account already exists on the machine on which you are installing the relay, you are prompted to enter the password. Enter the password and then click **Next**.
 - If a TFSRsrvd account does not exist, you are prompted to create an account:
 - a. Enter a password for this account in the **Password** field.
 - b. Enter the password again in the **Verify** field.
 - c. Click **Next**.
7. On the next dialog, enter the installation directory and click **Next**. The dialog for Relay-Parent Connection Options is displayed.



8. Complete the following fields regarding the connection between the relay and a parent machine, which can be either another relay or the endpoint proxy:

Relay Port

Enter the port number for the relay to communicate with the parent machine.

Relay or Endpoint Proxy Hostname

Enter the host name for the parent relay or endpoint proxy with which the relay will communicate.

Relay or Endpoint Proxy Port

Enter the port number for the parent relay or endpoint proxy with which the relay will communicate.

Click **Next**. The dialog for the type of relay-parent proxy connection is displayed.



9. Specify how the relay connects to the parent relay or endpoint proxy:
 - Select **Bidirectional** to permit connections that are initiated by either machine.
 - Select **Unidirectional** to permit connections initiated by only one machine. If you select this option, the **Relay Role** box is enabled:

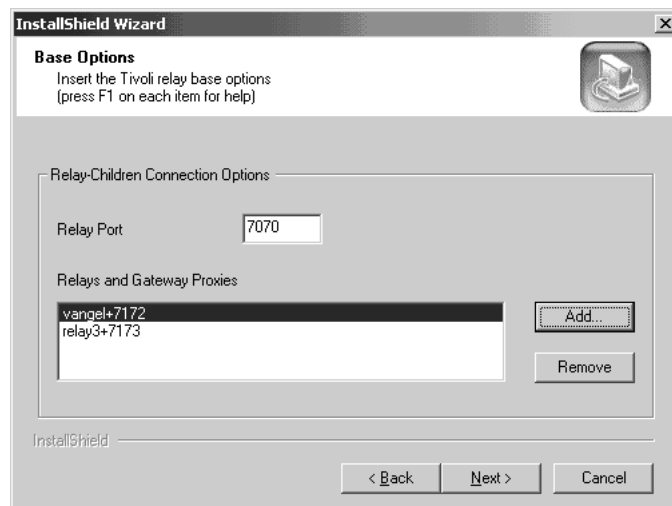
Initiator

The relay machine can start the connection with the parent machine.

Listener

The parent machine can start the connection with the relay machine.

Click **Next**. The dialog for relay-child connection options is displayed.



10. Complete the following fields:

Relay Port

Enter the port number for the relay to communicate with the children machines.

Relays and Gateway Proxies

Lists the relays and gateway proxies with which the relay connects.

To add a relay or gateway proxy to the list of destination machines, click **Add**. The Add Relay or Gateway Proxy dialog is displayed.



Complete the fields and click **OK**:

Hostname

Enter the host name of the child machine relay or gateway proxy with which the relay connects.

Port Enter the port number of the machine with which the relay connects.

To remove a machine, select it and click **Remove**.

Click **Next**. The dialog for the type of relay-child connection is displayed.



11. Specify how the relay connects to its children (relays and gateway proxies):
 - Select **Bidirectional** to permit connections that are initiated by either machine.
 - Select **Unidirectional** to permit connections initiated by only one machine. If you select this option, the **Relay Role** box is enabled:

Initiator

The relay machine can start the connection with the child machines.

Listener

The child machines can start the connection with the relay machine.

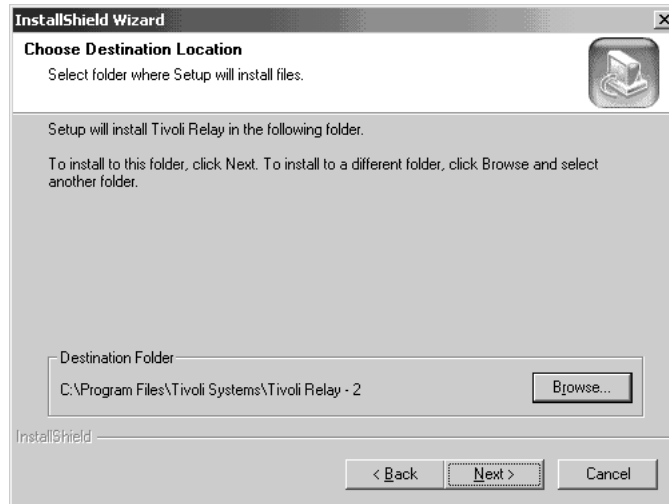
Click **Next**. The dialog shows a summary of your input.

12. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the relay.
13. A message asks you whether or not you want to start the relay. Click **Yes** to start it. Otherwise, click **No**. To start the component at another time, see “Starting and stopping the components” on page 53.

Installing additional relays on the same machine

To install additional instances, do the following:

1. From the directory that contains the Tivoli Relay installation images, double-click the setup.exe file. InstallShield Wizard starts. Select **Install** and click **Next**. The Choose Destination Location dialog is displayed.

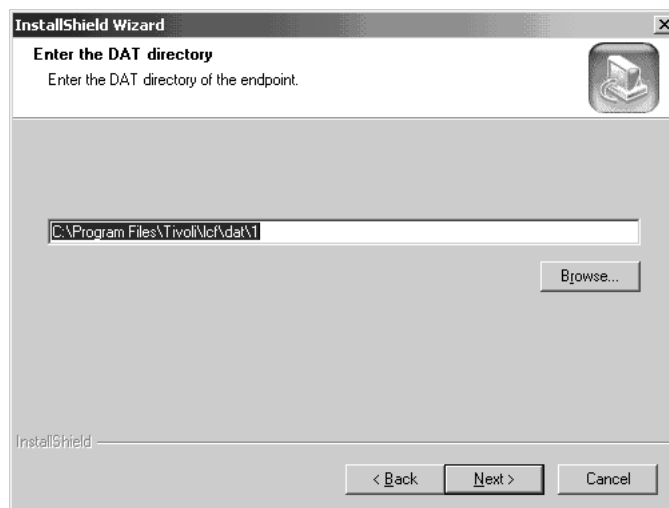


2. Enter the installation directory. Note that the default directory ends with a folder named for the numbered instance of the relay. This numbered name identifies this instance of the relay when you want to start, stop, and uninstall it.
If you change the default directory, ensure that it is different from the directory where you have installed previous instances of the relay.
Click **Next**.
3. Continue with step 8 on page 21 in the previous procedure.

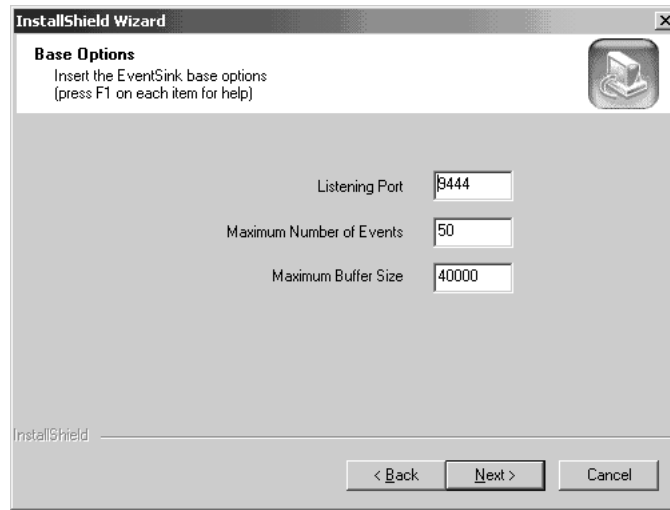
Installing a Windows event sink

You must install the event sink on an endpoint. To install the event sink, perform the steps, below. The event sink must be installed on a 4.3.1 endpoint:

1. From the directory that contains the Event Sink\w32-ix86\ subdirectory, double-click the Tivoli EventSink.exe file. The Tivoli Event Sink InstallShield Wizard starts.
2. Click **Next**.
3. On the next dialog, click **Yes** to accept the license agreement. The Destination Location dialog is displayed.



4. Specify the lcf_datdir directory of the endpoint on which you are installing the event sink. The eventsink.exe binary is installed in the lcf_datdir\..\..\bin\w32-ix86\mrt\ directory. The event sink configuration file is installed in the lcf_datdir directory. Click **Next**. The dialog for Event Sink Options is displayed.



5. Complete the following fields:

Port Enter the port number on the endpoint where the event sink will receive events. The default is 9444.

Maximum Number of Events

Enter the maximum number of events that the event sink will send to the Tivoli Enterprise Console server in a single package. The default is 50.

Maximum Buffer Size

Enter the maximum buffer size, in bytes, of the package that the event sink will send to the Tivoli Enterprise Console server. The default is 40000.

Click **Next**. The next dialog shows a summary of your input.

6. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the event sink.

Upgrading the components

The following sections describe how to upgrade the components from a previous version to the current version. There is no compatibility between Version 1.2 and Version 1.3.2 of Firewall Security Toolbox. When a component from Version 1.3.2 detects a connecting component from Version 1.2, it rejects the connection and logs the following error message:

```
ERROR multiplex.parseFrame: connecting peer is version 1.2.0
```

Version 1.3.1 and Version 1.3.2 of Firewall Security Toolbox are compatible.

Upgrading on Windows operating systems

To upgrade the component on a Windows machine, start the installation for the new version. The configuration and all the data of the previous version are saved and used by the new version.

Upgrading on UNIX operating systems

To upgrade the component on a UNIX machine, run the **upgrade.sh** script from the directory that contains the installation images.

If you are upgrading from Version 1.2, you are asked to enter the directory in which the previous version is installed. The configuration and all the data of the previous version are backed up to the directory that you specify with a suffix of **.1.2**. For example, if you specify `/usr/epp`, the backup directory becomes:
`/usr/epp.1.2`

If you are upgrading from Version 1.3.1, the upgrade script automatically detects the installation directory of the previous version. The configuration and all the data of the previous version are backed up to a directory with the same name as the current install directory plus a suffix that indicates the previously installed version. For example, if the installed version is 1.3.1-TFS-0002, and it is installed in `/usr/epp`, it will be backed up to a directory named `/usr/epp.1.3.1-TFS-0002`.

Uninstalling the components

The following sections describe how to uninstall the components.

Note: Under normal circumstances, you should not delete and reinstall an endpoint proxy. If the endpoint proxy is removed, all the dynamic configuration maintained in the `eproxy.bdb` file is also removed and lost. The reinstallation of the endpoint proxy cannot restore this information that is created during initial logins of endpoints. When you remove and reinstall the endpoint proxy, all endpoints that are connected to the endpoint proxy must do an initial login to the Endpoint Manager database as if they were new endpoints.

Make a backup copy of the `eproxy.bdb` file before uninstalling the endpoint proxy. After you reinstall the endpoint proxy, you can replace the `eproxy.bdb` file with your backup copy of the file. If you reinstall from the beginning, then you do not need to make a backup copy of the file. For more information, see “Backing up and restoring the endpoint manager database” on page 57.

The installation files are not removed when you uninstall the components. If you want to remove them, delete the files by hand.

Uninstalling from UNIX operating systems

To uninstall the component from a UNIX machine, run the **uninstall.sh** script from the directory in which the component is installed.

Uninstalling all components except the relay from Windows operating systems

You can uninstall all the components except the relay from a Windows system in one of the following ways:

- If you are using the disk image, run **setup.exe** and then choose **Remove**.
- If you installed the component using InstallShield, double-click **Add/Remove Programs** from the **Control Panel**. Select the component to remove, for example, Tivoli Endpoint Proxy, from the list of currently installed programs, and click **Add/Remove** or **Change/Remove**.

- Start the InstallShield Wizard that you used to install. Select **Remove** and click **Next**.

Note: If you are uninstalling one or more instances of a relay from the same machine, use this method.

In addition, delete the CFG and LOG files from the directory in which the component is installed. Table 2 lists the name of each file for each component:

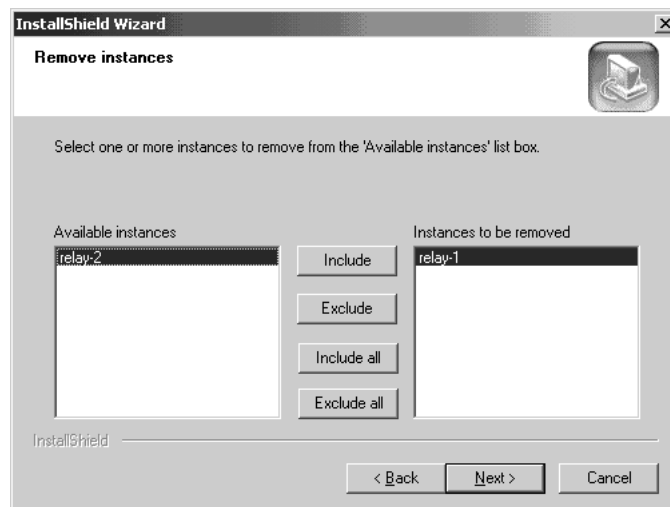
Table 2. CFG and LOG files by component

Endpoint Proxy	Event Sink	Gateway Proxy	Relay
epproxy.cfg	eventsink.cfg	gwproxy.cfg	relay.cfg
epp.log	eventsink.log	gwp.log	relay.log

Uninstalling the relay from Windows operating systems

To uninstall one or more instances of a relay from a Windows machine, perform the following steps:

1. Start the InstallShield Wizard that you used to install it. Select **Remove** and click **Next**. The Remove instances dialog is displayed. It lists the numbered relay instances that are installed on the machine.



2. From the Available instances list, select the instances that you want to remove and click **Include**, or click **Include all** to remove all the instances from the machine.
3. Click **Next** and confirm the deletion. The selected instances are deleted.

Note: The number of an instance that was uninstalled is recycled when you reinstall it. For example, if you uninstall relay-2, the next time you install an instance on the machine, the new instance is named relay-2.

Chapter 3. Configuring the components

This chapter explains how to configure the components of Firewall Security Toolbox.

For more information about configuring ports for each component of Firewall Security Toolbox, see Appendix C, “Configuring ports for Firewall Security Toolbox,” on page 77.

Configuring the endpoint proxy

After you install the endpoint proxy, the configuration file `eproxy.cfg` is created in the folder in which you installed the proxy. It contains the configuration input that you supplied during installation. In addition, to configure other options, edit the `eproxy.cfg` file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

keyword=value

The section titles are case-sensitive.

[endpoint-proxy] section

The [endpoint-proxy] section lists the main options for the endpoint proxy. Table 3 lists the keywords and a description.

Table 3. The endpoint-proxy section

Keyword	Description
gateway-host	The address and port number of the gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. The default port is 9494 and should not be changed unless the gateway is known to be using a different listening port with the endpoint. Use the format: <i>address+port</i>
gateway-interface	This option is used for multihomed endpoint proxies. It is the Domain Name System (DNS) name or IP address of the network interface of the endpoint proxies that is used to communicate with the gateway network interface.
accept-timeout	The timeout interval, in seconds, that the endpoint proxy waits for connections expected by the gateway. The default is 300.
max-sessions	The number of connections that the endpoint proxy can manage at the same time. The default is 75.
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the proxy and children or parent. This timeout ensures that the proxy cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240.

Table 3. The endpoint-proxy section (continued)

Keyword	Description
gateway-connection-timeout	The timeout interval, in seconds, for the endpoint proxy to wait for notification from an endpoint that it has completed sending data to the gateway before closing the connection to the gateway. This timeout ensures that the connection between the endpoint proxy and the gateway is closed. The default is (tcpip-timeout + 30) seconds. A value of 0 disables this feature.
port-range	The port ranges to use when allocating endpoint ports and when connecting with the gateway. For example, ports 7060, 8000 to 8050, 9000, and 9050 to 9080: 7060,8000-8050,9000,9050-9080 The default is 7000-9000.
database-path	The full path to a directory where the endpoint proxy database is installed. The endpoint proxy database records information about endpoints that it manages, and it must remain in existence even when the endpoint proxy is restarted. The default is the directory where the component is installed.
disable-udp	Enables the endpoint proxy to forward login requests via TCP instead of datagram protocol (UDP). To enable TCP, specify 1. To enable UDP, specify 0. The default is 0.
gateway-recv-buf-max	The size of the buffer, in KB, that the endpoint proxy uses to receive data from the gateway. The default value is 128.
resolve-address-retries	The number of times that an endpoint proxy retries an attempt to resolve a hostname. A value of zero means that the endpoint proxy makes one attempt to resolve the host name and performs no retries. The default value is 5.
resolve-address-delay	The time, in seconds, between attempts to resolve host names. A value of zero means that there is no delay between attempts. The default value is zero.
announce-interval	The interval, in seconds, between handshake signals sent by the endpoint proxy to its children. The default is 300 seconds. If this value is set to 0, the handshake signal is sent only once.
announce-always	Specifies whether the endpoint proxy continues to send handshake signals to the children after communication is established. When this parameter is set to 1, the endpoint proxy sends a handshake signal every <i>announce-interval</i> seconds, regardless of whether communication with children has been established. When this parameter is set to zero, the endpoint proxy stops sending handshake signals after communication is established. The default is zero.

[log] section

The [log] section lists log options. Table 4 on page 31 lists the keywords and a description.

Table 4. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by customer support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where endpoint proxy messages are written. The installation default is <code>epp.log</code> . If no file specified, the default is standard error. Note: Do not name the endpoint proxy log file <code>epproxy.log</code> . The name <code>epproxy.log</code> is reserved for the endpoint proxy database transaction file. If you specify <code>epproxy.log</code> as the value for <code>log-file</code> , the endpoint proxy will be unable to start.
max-size	The maximum size, in megabytes, that the log file can reach. When the log file reaches the maximum size, it is renamed to <code>filename.log.bak</code> and a new log file is started. For no limit, specify 0. The default is 1.

[communication-layer] section

The [communication-layer] section lists options for how the endpoint proxy connects to its relays or gateway proxies. Table 5 lists the keywords and a description.

Table 5. The communication-layer section

Keyword	Description
children-local-host	The network interface (DNS name or IP address) on the endpoint proxy to listen for communication from its relays or gateway proxies.
children-local-port	The port number of the endpoint proxy machine from which it listens for connections with relays or gateway proxies.
children-remote-list	The list of children hosts (relays or gateway proxies) to which the endpoint proxy connects. Separate the entries with a semicolon (;) but leave the end of the line without the delimiter. For example, a relay with address 69.99.99.71 and port 7071 and a gateway proxy with address 69.99.99.80 and port 7073: 69.99.99.71+7071;69.99.99.80+7073
children-remote-file	The name of the optional file containing a list of children hosts (relays or gateway proxies) to which the endpoint proxy connects. This keyword can be listed in addition to or instead of the <code>children-remote-list</code> keyword.
children-cm-type	The communication interface (TCP/IP) and the connectivity (unidirectional or bidirectional) that the endpoint proxy uses with its relays or gateway proxies. The values are <code>cm-tcp-bidirectional</code> , <code>cm-tcp-unidirectional</code> .

Table 5. The communication-layer section (continued)

Keyword	Description
enable-identity	<p>In a configuration in which Dynamic Network Address Translation (NAT) is used, the IP addresses of clients can change. This parameter allows support for dynamic Network Address Translation with unidirectional communications. Do not use this parameter when configuring bidirectional communications.</p> <p>Normally, when a component (a gateway proxy, endpoint proxy, or relay) is configured to use unidirectional communication, the server checks a client's IP address against the server's list of clients. If the address is not found, the connection is closed. If the address is found, the connection is associated with that client and communication is enabled.</p> <p>Setting enable-identity=1 allows a unidirectional client to send an "identity" packet to its peer server, and the server then uses the "identity," not the client's IP address to verify the client. This setting also changes the values used for other configuration file parameters in the other components. See "Configuring enable-identity" on page 49 for additional information.</p>
enable-tcp-nodelay	Determines whether the TCP/IP socket parameter TCP_NODELAY is enabled or disabled. Valid values are 0 and 1. Specifying 0 disables TCP_NODELAY; specifying 1 enables it. The default value is 0.
tcp-buffer-size	Specifies the buffer size, in KB, that TCP/IP uses to send and receive data. The default value is 0, which indicates that the system buffer size will be used.

Notes:

1. You must include at least one children host, specified either in the children-remote-list keyword or in the file listed for the children-remote-file keyword.
2. In the file specified for the children-remote-file keyword, list the children hosts on separate lines in the *host_name+port* format. Precede comments in the file with the number sign (#). You can leave blank lines as shown in the following example:

```
#Add these children hosts
```

```
luna+10023
sol+29993
```

[children-cm-info] section

The [children-cm-info] section lists further options about connectivity between the endpoint proxy and its children (relays or gateway proxies). Table 6 on page 33 lists the keywords and a description.

Table 6. The children-cm-info section

Keyword	Description
connection-mode	The role of the endpoint proxy in unidirectional connections only. The values are initiator or listener. The default is listener. Note: The values from Version 1.2 client (initiator) and server (listener), are still valid values.
local-port-range	The range of local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: 6060,7000-7070,9050,8000-8080 There is no default value for this parameter. If you do not specify a value, the operating system chooses the ports used to connect to the Firewall Security Toolbox child components.
receive-buffer-size	The size (in kilobytes) of the buffer that is used to receive communications from a TCP/IP socket. The minimum value is 1. The default is 17.
connect-timeout	The timeout interval, in seconds, after which a TCP/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0. The default value is zero. Note: If the endpoint proxy is configured as a unidirectional server, the default value is 30 seconds. You cannot set connect-timeout for unidirectional servers to 0.
send-timeout	The timeout interval, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, the number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. This option is the interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
polling-timeout	The timeout interval, in seconds, that a unidirectional initiator will wait before closing a connection with a unidirectional server. The default value is 240 seconds. polling-timeout should be set to a value that is larger than the drop-interval of the corresponding unidirectional server component. polling-timeout is ignored if specified for a component configured to communicate bidirectionally or as a unidirectional server.
drop-interval	In unidirectional connections for listener components only, the interval, in seconds, after which the listener suspends an inactive connection. The default is 5.

Table 6. The *children-cm-info* section (continued)

Keyword	Description
force-bind-address	This option is used for multi-homed endpoint proxies. It is the DNS name or IP address of the network interface of the endpoint proxy that is used to initiate connections with the children of the endpoint proxy.

Configuring the gateway proxy

After you install the gateway proxy, the `gwproxy.cfg` configuration file is created in the folder in which you installed the proxy. It contains the configuration input that you supplied during installation. In addition, to configure other options, edit the `gwproxy.cfg` file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

keyword=value

The section titles are case-sensitive.

[gateway-proxy] section

The [gateway-proxy] section lists the main options for the gateway proxy. Table 7 lists the keywords and a description.

Table 7. The *gateway-proxy* section

Keyword	Description
gateway-port	The port number on the gateway proxy that represents the gateway to the endpoints. The default is 9494.
gateway-interface	This option is used for multihomed gateway proxies. It is the DNS name or IP address of the gateway proxies network interface used to communicate with the Tivoli endpoints.
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the proxy and parent or endpoints. This timeout ensures that the proxy cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240.
endpoint-connection-timeout	The timeout interval, in seconds, for the gateway proxy to wait for notification from the gateway that it has completed sending data to a particular endpoint before closing the connection to that endpoint. This timeout ensures that the connection between the endpoint and the gateway proxy is closed. The default is (tcpip-timeout + 30) seconds. A value of 0 disables this feature.
proxy-label	Optional name to identify the gateway proxy instance. The default is the <i>host_name</i> .

Table 7. The gateway-proxy section (continued)

Keyword	Description
max-sessions	Number of connections that the gateway proxy can manage at the same time. The default is 75.
port-range	<p>The port ranges to use when connecting with endpoints. For example, ports 7060, 8000 to 8050, 9000, and 9050 to 9080:</p> <p>7060,8000-8050,9000,9050-9080</p> <p>The default is 7000-9000.</p>
reconnect-delay	The interval, in seconds, after which the gateway proxy attempts to reconnect to upstream components after a failed upcall. If an upstream component (endpoint proxy, relay, or gateway) is not reachable, the gateway proxy stops listening for endpoint-initiated connections. The gateway proxy then checks the status of the upstream components at the interval set by the reconnect-delay parameter. When connectivity to all upstream components can be reestablished, the gateway proxy resumes listening for endpoint-initiated connections. The default value for this parameter is 30 seconds.
resolve-address-retries	The number of times that a gateway proxy retries an attempt to resolve a hostname. A value of zero means that the gateway proxy makes one attempt to resolve the host name and performs no retries. The default value is 5.
resolve-address-delay	The time, in seconds, between attempts to resolve host names. A value of zero means that there is no delay between attempts. The default value is zero.
announce-interval	The interval, in seconds, between handshake signals sent by the gateway proxy to the parent. The default is 300 seconds. If this value is set to 0, the handshake signal is sent only once.
announce-always	Specifies whether the gateway proxy continues to send handshake signals to the parent after communication is established. When this parameter is set to 1, the gateway proxy sends a handshake signal every <i>announce-interval</i> seconds, regardless of whether communication with the parent has been established. When this parameter is set to zero, the gateway proxy stops sending handshake signals after communication is established. The default is zero.

[log] section

The [log] section lists log options. Table 8 on page 36 lists the keywords and a description.

Table 8. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by IBM Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where gateway proxy messages are written. The installation default is gwp.log. If no file specified, the default is standard error.
max-size	The maximum size, in megabytes, that the log file can reach. When the log file reaches the maximum size, it is renamed to <i>filename.log.bak</i> and a new log file is started. For no limit, specify 0. The default is 1.

[communication-layer] section

The [communication-layer] section lists options for how the gateway proxy connects to its relay or endpoint proxy. Table 9 lists the keywords and a description.

Table 9. The communication-layer section

Keyword	Description
parent-local-host	The DNS name or IP address of the gateway proxy from which it listens for connections from its relay or endpoint proxy.
parent-local-port	The port number of the gateway proxy machine from which it listens for connections with relay or endpoint proxy.
parent-remote-host	The DNS name or IP address of the relay or endpoint proxy.
parent-remote-port	The port number of the parent relay or endpoint proxy from which it listens for connections with gateway proxy.
parent-cm-type	The communication interface and the connectivity that the gateway proxy uses with its relay or endpoint proxy. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.
parent-remote-aliases	The list of alternate parent interfaces from which the gateway proxy can accept connections. This parameter is used when the parent host has multiple network interface cards and can use more than one interface to communicate with the gateway proxy. Entries are specified in the format <i>hostname+port</i> or <i>IP_address+port</i> . Separate the entries with a semicolon, as shown in the following example: parent-remote-aliases=sun+8012;9.17.44.22+7777

Table 9. The communication-layer section (continued)

Keyword	Description
enable-identity	<p>In a configuration in which Dynamic Network Address Translation (NAT) is used, the IP addresses of clients can change. This parameter allows support for dynamic Network Address Translation with unidirectional communications. Do not use this parameter when configuring bidirectional communications.</p> <p>Normally, when a component (a gateway proxy, endpoint proxy, or relay) is configured to use unidirectional communication, the server checks a client's IP address against the server's list of clients. If the address is not found, the connection is closed. If the address is found, the connection is associated with that client and communication is enabled.</p> <p>Setting enable-identity=1 allows a unidirectional client to send an "identity" packet to its peer server, and the server then uses the "identity," not the client's IP address to verify the client. This setting also changes the values used for other configuration file parameters in the other components. See "Configuring enable-identity" on page 49 for additional information.</p>
enable-tcp-nodelay	Determines whether the TCP/IP socket parameter TCP_NODELAY is enabled or disabled. Valid values are 0 and 1. Specifying 0 disables TCP_NODELAY; specifying 1 enables it. The default value is 0.
tcp-buffer-size	Specifies the buffer size, in KB, that TCP/IP uses to send and receive data. The default value is 0, which indicates that the system buffer size will be used.

[parent-cm-info]

The [parent-cm-info] section lists further options about connectivity between the gateway proxy and its parent (relay or endpoint proxy). Table 10 lists the keywords and a description.

Table 10. The parent-cm-info section

Keyword	Description
connection-mode	<p>The role of the gateway proxy in unidirectional connections only. The values are initiator or listener. The default is listener.</p> <p>Note: The values from Version 1.2 client (initiator) and server (listener) are still valid.</p>
local-port-range	<p>The range of the local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080:</p> <p>6060,7000-7070,9050,8000-8080</p> <p>There is no default value for this parameter. If you do not specify a value, the operating system chooses the port used to connect to the Firewall Security Toolbox parent component.</p>
receive-buffer-size	The size (in kilobytes) of the buffer that is used to receive communications from a TCP/IP socket. Minimum value: 1. The default is 17.

Table 10. The parent-cm-info section (continued)

Keyword	Description
connect-timeout	Timeout, in seconds, after which a TCP/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0. The default value is zero. Note: If the gateway proxy is configured as a unidirectional server, the default value is 30 seconds. You cannot set connect-timeout for unidirectional servers to 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. Interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
polling-timeout	The timeout interval, in seconds, that a unidirectional initiator will wait before closing a connection with a unidirectional server. The default value is 240 seconds. polling-timeout should be set to a value that is larger than the drop-interval of the corresponding unidirectional server component. polling-timeout is ignored if specified for a component configured to communicate bidirectionally or as a unidirectional server.
polling-timeout	The timeout interval, in seconds, that a unidirectional initiator will wait before closing a connection with a unidirectional server. The default value is 240 seconds. polling-timeout should be set to a value that is larger than the drop-interval of the corresponding unidirectional server component. polling-timeout is ignored if specified for a component configured to communicate bidirectionally or as a unidirectional server.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. The default is 5.
force-bind-address	This option is used for multi-homed gateway proxies. It is the DNS name or IP address of the network interface of the gateway proxy used to initiate connections with the parent of the gateway proxy.

Configuring the relay

After you install the relay, the configuration file relay.cfg is created in the folder in which you installed the component. It contains the configuration input that you supplied during installation. In addition, you can configure other options. To change these or configure other options, edit the relay.cfg file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

keyword=value

The section titles are case-sensitive.

[relay] section

The [relay] section is required at the top of the file, even when you do not specify any keywords. Table 11 lists the keyword and a description.

Table 11. The relay section

Keyword	Description
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the relay and children or parent. This timeout ensures that the relay cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240
resolve-address-retries	The number of times that a relay retries an attempt to resolve a hostname. A value of zero means that the relay makes one attempt to resolve the host name and performs no retries. The default value is 5.
resolve-address-delay	The time, in seconds, between attempts to resolve host names. A value of zero means that there is no delay between attempts. The default value is zero.
announce-interval	The interval, in seconds, between handshake signals sent by the relay to its children or parent. The default is 300 seconds. If this value is set to 0, the handshake signal is sent only once.
announce-always	Specifies whether the relay continues to send handshake signals to children or parent after communication is established. When this parameter is set to 1, the relay sends a handshake signal every <i>announce-interval</i> seconds, regardless of whether communication has been established. When this parameter is set to zero, the relay stops sending handshake signals after communication is established. The default is zero.

[log] section

The [log] section lists log options. Table 12 lists the keywords and a description.

Table 12. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by IBM Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where relay messages are written. Installation The default is relay.log. If no file specified, the default is standard error.

Table 12. The log section (continued)

Keyword	Description
max-size	The maximum size, in megabytes, that the log file can reach. When the log file reaches the maximum size, it is renamed to <i>filename.log.bak</i> and a new log file is started. For no limit, specify 0. The default is 1.

[communication-layer] section

The [communication-layer] section lists options for how the relay connects to its parent and children, relays, endpoint proxy, or gateway proxy. Table 13 lists the keywords and a description.

Table 13. The communication-layer section

Keyword	Description
parent-local-host	The DNS name or IP address of relay from which it listens for connections from its parent relay or endpoint proxy.
parent-local-port	The port number of the relay machine from which it listens for connections with parent relay or endpoint proxy.
parent-remote-host	The DNS name or IP address of the parent relay or endpoint proxy.
parent-remote-port	The port number of the parent relay or endpoint proxy from which it listens for connections with relay.
parent-cm-type	Communication interface and the connectivity that the relay uses with its parent relay or endpoint proxy. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.
children-local-host	DNS name or IP address of relay from which it listens for connections from children relays or gateway proxies.
children-local-port	The port the relay listens on for traffic from children relays or gateway proxies.
children-remote-list	List of children hosts (relays or gateway proxies) to which the relay connects. Separate the entries with a semicolon (;) but leave the end of the line without the delimiter. For example, a relay with address 69.99.99.71 and port 7071 and a gateway proxy with address 69.99.99.80 and port 7073: 69.99.99.71+7071;69.99.99.80+7073
children-cm-type	Communication interface and the connectivity that the relay uses with its relays or gateway proxies. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.
children-remote-file	The name of the optional file containing a list of children hosts (relays or gateway proxies) to which the relay connects. This keyword can be listed in addition to or instead of the children-remote-list keyword.
parent-remote-aliases	The list of alternate parent interfaces from which the relay can accept connections. This parameter is used when the parent host has multiple network interface cards and can use more than one interface to communicate with the relay. Entries are specified in the format <i>hostname+port</i> or <i>IP_address+port</i> . Separate the entries with a semicolon, as shown in the following example: parent-remote-aliases=sun+8012;9.17.44.22+7777

Table 13. The communication-layer section (continued)

Keyword	Description
enable-identity	<p>In a configuration in which Dynamic Network Address Translation (NAT) is used, the IP addresses of clients can change. This parameter allows support for dynamic Network Address Translation with unidirectional communications. Do not use this parameter when configuring bidirectional communications.</p> <p>Normally, when a component (a gateway proxy, endpoint proxy, or relay) is configured to use unidirectional communication, the server checks a client's IP address against the server's list of clients. If the address is not found, the connection is closed. If the address is found, the connection is associated with that client and communication is enabled.</p> <p>Setting enable-identity=1 allows a unidirectional client to send an "identity" packet to its peer server, and the server then uses the "identity," not the client's IP address to verify the client. This setting also changes the values used for other configuration file parameters in the other components. See "Configuring enable-identity" on page 49 for additional information.</p>
enable-tcp-nodelay	Determines whether the TCP/IP socket parameter TCP_NODELAY is enabled or disabled. Valid values are 0 and 1. Specifying 0 disables TCP_NODELAY; specifying 1 enables it. The default value is 0.
tcp-buffer-size	Specifies the buffer size, in KB, that TCP/IP uses to send and receive data. The default value is 0, which indicates that the system buffer size will be used.

Notes:

1. You must include at least one children host, specified either in the children-remote-list keyword or in the file listed for the children-remote-file keyword.
2. In the file specified for the children-remote-file keyword, list the children hosts on separate lines in the *host_name+port* format. Precede comments in the file with the number sign (#). You can leave blank lines as shown in the following example:

```
#Add these children hosts

luna+10023
sol+29993
```

[children-cm-info] section

The [children-cm-info] section lists further options about connectivity between the relay and its children (relays or gateway proxies). Table 14 on page 42 lists the keywords and a description.

Table 14. The children-cm-info section

Keyword	Description
connection-mode	The role of relay in unidirectional connections only. The values are initiator or listener. The default is listener. Note: The values from Version 1.2 client (initiator) and server (listener) are still valid.
local-port-range	The range of local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: 6060,7000-7070,9050,8000-8080
receive-buffer-size	The size (in kilobytes) of the buffer that is used to receive communications from a TCP/IP socket. The minimum value is 1. The default is 17.
connect-timeout	The timeout interval, in seconds, after which a TCP/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0. The default value is zero. Note: If the relay is configured as a unidirectional server, the default value is 30 seconds. You cannot set connect-timeout for unidirectional servers to 0.
send-timeout	The timeout interval, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. Interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
polling-timeout	The timeout interval, in seconds, that a unidirectional initiator will wait before closing a connection with a unidirectional server. The default value is 240 seconds. polling-timeout should be set to a value that is larger than the drop-interval of the corresponding unidirectional server component. polling-timeout is ignored if specified for a component configured to communicate bidirectionally or as a unidirectional server.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. The default is 5.
force-bind-address	This option is used for multi-homed relays. It is the DNS name or IP address of the network interface of the relay that is used to initiate connections with the children of the relay.

[parent-cm-info] section

The [parent-cm-info] section lists further options about connectivity between the relay and its parent (relay or endpoint proxy). Table 15 on page 43 lists the keywords and a description.

Table 15. The parent-cm-info section

Keyword	Description
connection-mode	Role of relay in unidirectional connections only. Values: initiator or listener. The default is listener. Note: The values client (initiator) and server (listener), which were accepted values for Version 1.2, remain valid.
local-port-range	Range of local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: 6060,7000-7070,9050,8000-8080
receive-buffer-size	Size of buffer, in kilobytes, used to receive from a TCP/IP socket. Minimum value: 1. The default is 17.
connect-timeout	Timeout, in seconds, after which a TCP/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0. The default value is zero. Note: If the relay is configured as a unidirectional server, the default value is 30 seconds. You cannot set connect-timeout for unidirectional servers to 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. Interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
polling-timeout	The timeout interval, in seconds, that a unidirectional initiator will wait before closing a connection with a unidirectional server. The default value is 240 seconds. polling-timeout should be set to a value that is larger than the drop-interval of the corresponding unidirectional server component. polling-timeout is ignored if specified for a component configured to communicate bidirectionally or as a unidirectional server.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. The default is 5.
force-bind-address	This option is used for multi-homed relays. It is the DNS name or IP address of the network interface of the relay that is used to initiate connections with the parent of the relay.

Configuring the event sink

After you install the event sink, the configuration file eventsink.cfg is created in the folder in which you installed the component. It contains the configuration input that you supply during installation.

Note: You must also configure every generator of non-secure events in your environment to send events to the event sink and not to the Tivoli Enterprise Console server. To configure non-TME adapters, see “Configuring non-TME adapters for the event sink” on page 46. To configure Availability Intermediate Manager Console server see “Processing events from the Tivoli Enterprise Console Availability Intermediate Manager console” on page 58.

To configure other options, edit the `eventsink.cfg` file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

`keyword=value`

The section titles are case-sensitive.

[SENDING] section

The [SENDING] section lists options for sending events to the Tivoli Enterprise Console server. Table 16 lists the keywords and a description.

Table 16. The *SENDING* section

Keyword	Description
<code>lcf-datdir</code>	The \dat directory of the endpoint.
<code>max-size-buffer</code>	The maximum buffer size, in bytes, of the package that the event sink sends to the Tivoli Enterprise Console server. The default is 40000.
<code>max-num-events-to-send</code>	The maximum number of events that the event sink sends to the Tivoli Enterprise Console server in a single package. The default is 50.
<code>delay-time</code>	The minimum interval, in seconds, between sending packages of events to the Tivoli Enterprise Console server. To send packages immediately, specify 0. The default is 1.
<code>caching-timeout</code>	The timeout interval, in seconds, by which the event sink sends events if neither the maximum buffer size nor the maximum number of events is reached. The default is 30.

[RECEPTION] section

The [RECEPTION] section lists options for receiving events from non-TME adapters. Table 17 lists the keywords and a description.

Table 17. The *RECEPTION* section

Keyword	Description
<code>port</code>	The port number on the endpoint where the event sink receives events. The default is 9444.
<code>max-sessions</code>	The maximum number of threads that the event sink can have with the non-TME adapters. Set this value at least to equal the number of non-TME adapters with which the event sink communicates. The default is 100.

Table 17. The *RECEPTION* section (continued)

Keyword	Description
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the event sink and non-TME adapters. This timeout ensures that the event sink cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240.
max-ram-cache	The amount of memory, RAM, in kilobytes, in which events get held on the event sink machine. The event sink stops receiving events when this value is reached, until it sends the events to the Tivoli Enterprise Console server. The default is 1024.
caching-timeout	The timeout value, in seconds, by which the event sink sends events if neither the maximum buffer size nor the maximum number of events is reached. The default is 30.

[EIF] section

The [EIF] section lists options for the Tivoli Enterprise Integration Facility.

The event sink is a Tivoli Enterprise Console endpoint adapter. That is, the event sink uses TEC methods to send events upstream. The TEC methods send the events to the lcf process (i.e. the locally installed endpoint). The endpoint sends the events to a TEC gateway, which in turn forwards the events to an event server.

Table 18 lists only the keywords and a description for the parameters that are required for Firewall Security Toolbox.

Table 18. The *EIF* section

Keyword	Description
BufEvtMaxSize	The maximum size, in kilobytes, of the eventsink.cache file. The default is 64.
BufEvtPath	The file in which the event sink saves events that it temporarily cannot send. By default, this file is created in the installation directory on UNIX and in the dat directory on Windows machines. The default is eventsink.cache.
ServerLocation	Optional. Specifies the resource name of the Tivoli event server that the event sink is to use. To forward events through the normal Tivoli channels, use the default value. The default is @EventServer. Use the following syntax: ServerLocation=@EventServer#region_name

[LOG] section

The [log] section lists log options. Table 19 on page 46 lists the keywords and a description.

Table 19. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by customer support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where event sink messages are written. The default for Windows operating systems is: <i>/DAT_directory/eventsink.log</i> . The default for UNIX operating systems is: <i>/installation_directory/eventsink.log</i> . If no file is specified, the default is standard error.
max-size	The maximum size, in megabytes, that the log file can reach. For no limit, specify 0. The default is 1.

Configuring non-TME adapters for the event sink

To configure the non-TME adapter to send events to the event sink and not to the Tivoli Enterprise Console server, edit the configuration file on the non-TME adapter and change the following parameters:

```
ServerLocation=host_name  
ServerPort=port
```

Where:

host_name

The host name of the endpoint on which the event sink is installed

port

The port on which the event sink listens for events.

Migrating endpoints to connect to a gateway proxy

To migrate an endpoint from a Tivoli gateway to a gateway proxy, change the login interfaces of the endpoint by specifying gateway proxies in the list of interfaces. Do one of the following:

- Use the **wep set interfaces** command:

1. From the Tivoli server or managed node, enter the command:

```
wep set interfaces -e ep_label host_name_gwp+port
```

where:

ep_label

Is the label of the endpoint

host_name_gwp

Is the host name of the gateway proxy

port

Is the port number of the gateway proxy

2. Enter the command:

```
wep sync_gateways
```

3. Stop the endpoint.

4. Put up the firewall between the endpoint and the gateway that managed it previously.

5. Restart the endpoint.

- Use the HTTP interface of the endpoint:
 1. Update the login interfaces and gateway to point to one or more gateway proxies.
 2. Use a Web browser to update each endpoint.
 3. From the Network Address Configuration menu, use the **lcmd -g** command to set the gateway. Additionally or alternatively, use the **-D lcs.login_interfaces** option. Note that you must know the HTTP user name and password for the endpoint. See the *Tivoli Management Framework Maintenance and Troubleshooting Guide* for details.
 4. Put up the firewall between the endpoint and the gateway that managed it previously.
 5. Restart the endpoint.

The following commands and scripts are *not* supported for endpoints and gateways that have the firewall proxies between them:

- **wep migrate** command
- **select_gateway_policy** script

To migrate an endpoint from a gateway proxy to a Tivoli gateway, use the **wep migrate** command as you would normally.

Configuring backup gateway proxies

You can set up the components to include more than one gateway proxy in a DMZ so that if a gateway proxy is down, the endpoint proxy can use an alternative gateway proxy to reach an endpoint. This process of looking for an alternative gateway proxy is called a *gateway proxy failover*.

To set up alternative gateway proxies, you create groups of gateway proxies that the endpoint proxy tries to use in the order that you specify.

Do the following:

1. Create a file named `proxy.grp` in the directory where the endpoint proxy is installed. The account with which the endpoint proxy runs must have the permissions to read the file.
2. In the `proxy.grp` file, include a single line entry for each group of gateway proxies that you want to create. For example:


```
group1: a b c
group2: a d e f
```

Where `group1` and `group2` are the names of groups of gateway proxies. The letters `a` through `f` are the labels of gateway proxies.

Follow each group name with a colon (:). The group names can be whatever you like as long as each name is unique in the file.

List each gateway proxy in the order in which the endpoint proxy should search for it. Use the gateway proxy label that is specified either at installation or in the configuration file of the gateway proxy. If gateway proxy 'a' is down, the endpoint proxy tries 'b'. If 'b' is down, it tries 'c'.

You can specify the same gateway proxy in more than one group, for example, `a` in both `group1` and `group2`. When gateway proxy `a` fails, the endpoint proxy will try all the gateway proxies in the groups that contain gateway proxy `a`.

The **gateway-port** keyword must be the same for all gateway proxies that are listed in the same group in the proxy.grp file. Otherwise, the gateway proxy failover will not work for that group.

Configuring endpoints for backup gateway proxies

In addition to specifying the list of backup gateway proxies for the endpoint proxy, you must configure the endpoint to connect to a specific list of gateway proxies when the main gateway proxy is unavailable.

Gateway proxies are designed to migrate endpoints to backup gateway proxies when upcalls fail to reach upstream components. When an upcall from an endpoint is passed by the gateway proxy to upstream components and then the upcall fails, the gateway closes its port to the endpoint. The gateway proxy then polls the upstream components (at intervals determined by the reconnect-delay parameter) to determine when they become available. When the upstream components are available, the gateway proxy reopens the port to the endpoint. If the endpoint attempts another upcall while the port is closed, the endpoint fails to reach the main gateway proxy and migrates to a backup gateway proxy.

When the endpoint migrates to a backup gateway proxy, it will complete its upcall successfully only if the upcall can bypass the upstream component that blocked communication in the first place. Therefore, to maximize the benefit of endpoint migration, the backup gateway proxy should ideally connect to a different endpoint proxy and gateway than the main gateway proxy.

For example, an endpoint upcall fails because an endpoint proxy is down, so the endpoint migrates from the main gateway proxy to a backup gateway proxy. If the backup gateway proxy connects to the same endpoint proxy as the main gateway proxy, the upcall will still fail.

For unidirectional configurations in which the child is the server and the parent is the client, you must also configure the gateway proxy connection-timeout parameter to an appropriate value for the child to determine that the parent is unavailable. Set this parameter to a value that is larger than the value of the gateway proxy polling-interval parameter.

To configure the endpoint to connect to a specific list of gateway proxies, edit the **login_policy** script to configure it to work with Firewall Security Toolbox. Refer to *Tivoli Management Framework Reference Manual* for more information about the **login_policy** script.

Add the following to the **login_policy** script:

- List the gateway proxies for the endpoint in the following format:

`gateway_proxy1+port1:gateway_proxy2+port2`

where:

`gateway_proxy1+port1`

Indicates the host name or IP address of the main gateway proxy and its port number.

`gateway_proxy2+port2`

Indicates the host name or IP address of the alternative gateway proxy and its port number.

- The **wep set interfaces** command

For example:

```
wep set interfaces -e $1 gwp_list
```

where:

\$1 Is the endpoint label as defined in the login policy.

gwp_list

Is the list of gateway proxies as you defined them in your logic in the login policy.

If in a Tivoli region there are some endpoints that connect directly to a gateway because there is no firewall between them, and some endpoints that connect through Firewall Security Toolbox, these changes to the **login_policy** script must be run only for the endpoints that connect through Firewall Security Toolbox. This keeps endpoints that connect directly to a gateway from trying to connect through Firewall Security Toolbox.

To understand whether the endpoint is logging in through Firewall Security Toolbox, use the **login_policy** script in one of the following ways:

- Check the \$5 value, the IP address of the endpoint. If the IP address is the same as that of the endpoint proxy, the endpoint is logging in through the Firewall Security Toolbox.
- Reserve one gateway in the Tivoli region to log in endpoints through Firewall Security Toolbox. Check the \$4 value, the object reference of the gateway that intercepted the login request from the endpoint. If the value is the same as that of the reserved gateway, the endpoint is logging in through Firewall Security Toolbox.
- Use a naming convention for endpoint labels that identifies only endpoints that log in through Firewall Security Toolbox. Check the \$1 value, the endpoint label. If the value matches the naming convention, the endpoint is logging in through Firewall Security Toolbox.

For new endpoints: When you first install an endpoint, specify the gateway proxy and its ports.

The login policy that you defined will be run and the login interfaces will be updated to add backup gateways.

For existing endpoints: To start the login policy for the endpoints, enter the following command:

```
wadminep endpoint_label reexec_lcmd
```

where *endpoint_label* is the label of the endpoint.

Configuring enable-identity

If you have set enable-identity=1 in the [communication-layer] portion of any component's configuration file, the following configuration settings must also be changed:

If the client is a parent:

If the client (the initiator) is a parent component, the **parent-local-host** parameter is used to specify the identity value that the client sends to the server. The value that you specify is an IP address that represents the computer that the client runs on. The IP address specified does not have to

actually exist anywhere in the NAT environment, but it can be an actual IP address if you want it to be. The address specified, however, must be unique in the NAT environment.

Then, use the **children-remote-list** parameter in the server component configuration file to specify the client's identity value. The value to specify on **children-remote-list** is the IP address specified on the parent, plus an arbitrary port number (an integer).

If the client is a child:

If the client is a child component, the **children-local-host** parameter is used to specify the identity value that the client sends to the server. The value that you specify is an IP address that represents the computer that the client runs on. The IP address specified does not have to actually exist anywhere in the NAT environment, but it can be an actual IP address if you want it to be. The address specified, however, must be unique in the NAT environment.

Then, use the **parent-remote-host** parameter in the server component configuration file to specify the client's identity value. The value to specify on **parent-remote-host** is the IP address specified on the parent, a plus (+) sign, and any unused port number.

Example 1: Assume that an endpoint proxy, named **epp1**, is a unidirectional server, and its child, a gateway proxy named **gwp1** is a unidirectional client. Also assume that the actual IP address of the gateway proxy (**gwp1**) is 121.17.67.5, and that the identity value that you wish to assign to the gateway is 1.2.3.4.

The configuration files for both **epp1** (epproxy.cfg) and **gwp1** (gwproxy.cfg) would both need to have **enable-identity=1** in the [communication-layer] section.

Because **gwp1** is a child component of **epp1**, the gwproxy.cfg file for **gwp1** would also need to contain an entry for the identity value on the **parent-local-host** parameter. For example:

```
[communication-layer]
parent-local-host=1.2.3.4
```

Note: the IP address to be used as the identity does not have to be an actual IP address that exists in the NAT environment. Whatever you specify, it must be unique among all the other identity values that you use in other components in the NAT environment. It can be an existing IP address; but it does not have to be.

Because **epp1** is a parent to **gwp1**, the epproxy.cfg file for **epp1** would need to include an entry like the following to specify the address and include:

```
[communication-layer]
children-remote-list=1.2.3.4+1234
```

Note: The number specified (1234, in this example) does not have to be unique, but it should correspond to any unused port number.

On any relays that are being used for unidirectional communications:

In the next example, a relay component is installed between **epp1** and **gwp1**. **epp1** is the relay's parent. The relay acts as a client, and **epp1** is its server. **gwp1** is a child of the relay. The relay acts as a server for **gwp1**. Assume that the relay's

server IP address is 147.29.17.2, and its server port is 6804. The relay will use an “identity” value when it communicates with **epp1** and **gwp1** will use an “identity” value when it communicates with the relay.

- In relay.cfg, add **enable-identity=1**.
- In the epproxy.cfg file for **epp1**, a new identity value must be added for the relay, on the **children-remote-list** parameter. For example, set **children-remote-list=2.3.4.5+5432**.
- In relay.cfg, set **parent-local-host** and **children-host-list** as follows,
parent-local-host=2.3.4.5
children-remote-list=1.2.3.4+1234
- The gwproxy.cfg file for **gwp1** needs to set **parent-remote-host=147.29.17.2** (the relay’s actual IP address).
- In relay.cfg, enable unidirectional communications:
children-cm-type=cm-type-unidirectional
parent-cm-type=cm-type-unidirectional

Chapter 4. Using Firewall Security Toolbox

This chapter describes how to work with Firewall Security Toolbox in your environment.

Starting and stopping the components

If you did not start the components when you installed them, you need to start them to use Firewall Security Toolbox.

Starting and stopping the components on Windows operating systems

On Windows operating systems, do the following:

1. Open Services from the Control Panel.
2. Select the component from the list of services:
 - Tivoli Endpoint Proxy
 - Tivoli Event Sink
 - Tivoli Gateway Proxy
 - Tivoli Relay. If you have more than one instance of the relay installed on the machine, select the numbered instance that you want to start.
3. Select **Start** from the pop-up menu.

To stop the components on Windows operating systems, select **Stop** from the pop-up menu instead.

Starting and stopping the components on UNIX operating systems

On UNIX operating systems, to start the components, do the following:

1. Go to the directory in which the component is installed.
2. Enter the command:

```
./component.sh start
```

where *component* stands for:

epproxy

The endpoint proxy

eventsink

The event sink

gwproxy

The gateway proxy

relay

The relay. If you have more than one instance of the relay installed on the machine, specify the directory numbered for the instance of the relay to be started or stopped.

To stop the components on UNIX operating systems, enter the command from the directory in which the component is installed:

```
./component.sh stop
```

Working with endpoints logged in through the proxy

Note: Never modify the attributes of an endpoint, reassign an endpoint to a new endpoint proxy, or remove an endpoint from the database when the endpoint proxy is running. These actions can corrupt the endpoint proxy database file (epproxy.bdb). Instead make a copy of the database and edit the copy:

1. Copy epproxy.bdb to an editable location.
2. Edit the copy using **wproxy -d /someDirectoryName edit**.
3. Stop the endpoint proxy (**epproxy.sh stop** or **net stop endpointproxy**)
4. Copy the edited file back into the endpoint proxy directory.
5. Start the endpoint proxy daemon/service (**epproxy.sh start** or **net start endpointproxy**)

Endpoints that log in with the Tivoli server through the endpoint proxy are recorded in the endpoint proxy database (epproxy.bdb). To work with these endpoints, use the **wproxy** command. Before using the **wproxy** command, ensure the following:

- That you have logged on with the account with which the endpoint proxy runs. You must use the same account that was specified to run the endpoint proxy.
- That you have set up the shell environment by running `setup_env.sh` from the directory in which the endpoint proxy is installed.

Listing the endpoints in the database

To list the endpoints in the endpoint proxy database, enter the following command:

```
wproxy db [-d db_directory] ls [odnum...]
```

where *db_directory* indicates the directory in which the database is stored.

The results appear in the following format:

```
ep_label=label odnum=identifier address=IP_address proxy_port=port1  
real_port=port2 gwp_label=gateway_proxy
```

where:

ep_label

Indicates the label of the endpoint.

identifier

Indicates the number assigned to the endpoint by the Tivoli Management Framework.

IP_address

Indicates the address of the endpoint.

odnum

Indicates the number assigned to the endpoint by Tivoli Management Framework.

port1

Indicates the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. It is a port that the endpoint proxy uses to pose as the endpoint to the gateway.

port2

Indicates the endpoint port that the gateway proxy uses to communicate with the endpoint.

gateway_proxy

Indicates the label of the gateway proxy with which the endpoint connects.

See “wproxy” on page 62 for more details.

Modifying the attributes of an endpoint

You can modify the attributes of an endpoint, for example, the gateway proxy with which it connects.

To modify one or more attributes of an endpoint, enter the following command:

```
wproxy db edit odnum attribute=value...
```

where:

odnum Indicates the number assigned to the endpoint by Tivoli Management Framework.

attribute

Indicates the attributes that you can modify:

proxy_port

Indicates the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. It is a port that the endpoint proxy uses to pose as the endpoint to the gateway.

real_port

Indicates the endpoint port that the endpoint uses to listen for connections from the gateway proxy. It is the same port as the *lcfcd_port* in the *last.cfg* file.

gwp_label

Indicates the label of the gateway proxy with which the endpoint connects.

value Indicates the new value that you want to assign to the attribute.

See “wproxy” on page 62 for more details.

Reassigning an endpoint to a new gateway proxy

By modifying the gateway proxy label for an endpoint, you can reassign an endpoint to a new gateway proxy to balance the load on the gateway proxies in your configuration. Perform the following main steps:

1. Run the **wep set interface** command.
2. Run the **wproxy db edit** command (see “Modifying the attributes of an endpoint” for the syntax) and specify the new *gwp_label* value.
3. To update the gateway settings in the *lcf.dat* file of the endpoint, run the **wep ep_label status** command.

See “wproxy” on page 62 for more details.

Changing the port on the endpoint

If you change the port that is used by the endpoint to listen for connections from the gateway (for example, if you restarted the endpoint after editing the *lcfcd_port* value in the *last.cfg* file) the *real_port* value could get out of sync with the *lcfcd_port* value. You can bring the ports back in sync by modifying the *real_port* value.

Run the **wproxy db edit** command and specify the new *real_port* value.

See “wproxy” on page 62 for more details.

Reassigning an endpoint to a new endpoint proxy

You can use the **wproxy** command to move an endpoint from one endpoint proxy to another. To reassign an endpoint to a new endpoint proxy, perform the following steps:

1. To export the data of the endpoints that you want to move, enter the following command from the source endpoint proxy:

```
wproxy db export file_name odnum ...
```

where *file_name* is the name of the file to which the endpoint data is exported, and *odnum* is the number assigned to an endpoint by Tivoli Management Framework. You can specify one or more endpoints.

The endpoint data is exported in the format shown in the following example:

```
ep_label=ozark odnum=2 address=9.14.28.108 proxy_port=7001 real_port=10221  
gwp_label=ozark-gwp crypt_mode=0 crypt_key=eb31cb3e47f56844a44c4537ce8d654  
3a9aa8db09ae6fc47a0482db39717fc8c84452b291478dd35
```

2. To import the endpoint data, enter the following command from the destination endpoint proxy:

```
wproxy db import file_name
```

where *file_name* is the name of the file to which the endpoint data was exported in step 1.

3. Reassign the endpoint to a new gateway proxy using the following command:

```
wproxy db edit odnum gwp_label=label proxy_port=port
```

where *label* is the label of the gateway proxy with which the endpoint connects and *port* is the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. You must run this command separately for each endpoint.

4. Assign the endpoint to the new gateway proxy for normal logins. To do this, edit the following parameter in the last.cfg file of the endpoint:

```
lcs.gateway_address=hostname+port
```

where *hostname* is the host name or IP address of the new gateway proxy, and *port* is the port that the gateway proxy uses for communication with the endpoint.

5. Stop and then restart the endpoint service.

See “wproxy” on page 62 for more details about the **wproxy** command.

Removing an endpoint from the database

When endpoints are deleted from a Tivoli region, they are not automatically deleted from the endpoint proxy database. You must remove them manually. To remove one or more endpoints from the endpoint proxy database, enter the following command:

```
wproxy db -d db_directory remove odnum
```

where:

db_directory

Indicates the directory in which the database is stored.

odnum Indicates the number assigned to the endpoint by Tivoli Management Framework.

See “wproxy” on page 62 for more details.

Backing up and restoring the endpoint manager database

When you back up the endpoint manager database using the **wbkupdb** command, the endpoint proxy database is not backed up. To keep a version of the endpoint proxy database that reflects the state of the endpoints when you backed up the endpoint manager database, make a copy of the `epproxy.bdb` file and store it with the output file of the **wbkupdb** command. If you restore the endpoint manager database, stop the endpoint proxy, copy the backup `epproxy.bdb` file to the directory in which the endpoint proxy is installed, then restart the endpoint proxy.

Installing endpoints in a DMZ

The following sections describe how to install or configure endpoints in a DMZ.

Installing the endpoints from scratch

Installing UNIX endpoints using the **winstlcf** command across a firewall is supported only if the firewall allows `rexec` traffic to pass. This is probably not enabled in a production environment. Alternatively, you can use one of the following methods:

- Shut down the firewall for the time necessary to install and configure the endpoints using the **winstlcf** command, and then enable it again when the endpoints are ready to run.
- Open the firewall to permit the **winstlcf** command to be sent through the `rexec` port on UNIX endpoints.
- Create a separate Tivoli region in the DMZ for installing endpoints only. Shut down the Tivoli region after installation and leave it dormant unless you need to install other endpoints. Then, configure the endpoints to communicate with the gateway proxy.

Connecting endpoints that are already present in the Tivoli region

To connect endpoints that were connected to a gateway to a gateway proxy instead, do the following:

1. Delete the endpoint from the Tivoli region to which it is connected using the **wdelep** command.
2. Stop the endpoint.
3. Reconfigure the endpoint:
 - a. From the endpoint DAT directory, delete all *except* the following files:
 - `last.cfg`
 - `lcf_env.csh`
 - `lcf_env.sh`
 - `lcf.d.sh`
 - b. Edit the `last.cfg` file:
 - Change the **gateway_port** entry to the following:
`gateway_port=gateway_proxy_port`

where *gateway_proxy_port* is the port number of the gateway proxy to which you are migrating the endpoint.

- Add the following entry:

```
lcs.login_interfaces=gateway_proxy_host_name+port
```

where *gateway_proxy_host_name+port* is the host name and the port number of the gateway proxy.

4. Save and close the file, and restart the endpoint.

Existing installation methods that are not based on remote access work as well.

Processing events from the Tivoli Enterprise Console Availability Intermediate Manager console

If you use Tivoli Enterprise Console Availability Intermediate Manager console, you need to configure it to work with Firewall Security Toolbox. When there is a firewall between the machine with Tivoli Enterprise Console Availability Intermediate Manager console and the Tivoli Enterprise Console server and you need to send events to Tivoli Enterprise Console, you must send events to the event sink instead of to the Tivoli Enterprise Console server directly. The event sink then forwards the events to the Tivoli Enterprise Console server across the firewalls.

If Tivoli Enterprise Console Availability Intermediate Manager console is set up to send the events to the Tivoli Enterprise Console server to be processed, do the following:

1. Customize the action "Send a TEC Event to a TEC Server." The Customize Action dialog is displayed.
2. In the **IP Address or Hostname** text box, enter the host name or address of the event sink.
3. In the **Server Port** text box, enter the port of the event sink.
4. Click **Save Event Action**.
5. Distribute the Event Action Plan to the Tivoli Enterprise Console Availability Intermediate Manager.

If you have a rule base that processes the events on the Tivoli Enterprise Console Availability Intermediate Manager console and forwards them to the Tivoli Enterprise Console server, do the following:

1. In the `/dat/default_rb/TEC_Rules/` directory where the console is installed, edit the `tec_forward.conf` file:
 - For the `ServerLocation` entry, specify the host name of the event sink.
 - For the `ServerPort` entry, specify the port number of the event sink.
 - For the `TestMode` entry, specify **No**.

Save and close the file.

2. Customize the action "Send a Tivoli Enterprise Console Event to a Tivoli Enterprise Console Server." The Customize Action dialog is displayed.
3. In the **IP Address or Hostname** text box, enter the host name or address of the Tivoli Enterprise Console Availability Intermediate Manager.
4. In the **Server Port** text box, enter the port of the Tivoli Enterprise Console Availability Intermediate Manager.
5. Click **Save Event Action**.

6. Distribute the Event Action Plan to the Tivoli Enterprise Console Availability Intermediate Manager.

Viewing endpoint properties

You can view the properties of an endpoint using a Web browser by entering the host name and port number of the endpoint in the **Location** text box.

When the endpoint is connected to the Tivoli environment through Firewall Security Toolbox proxies, enter the following URL in the **Location** text box:

http://host_name:port_number

where:

host_name

Indicates the host name of the endpoint proxy, not the endpoint.

port_number

Indicates the port number that the endpoint proxy uses to pose as the endpoint. To get the port number, you can, for example, view the list of endpoints for the gateway from the gateway list and select the endpoint.

Appendix A. Using the command line interface

This appendix describes the **wproxy** command.

Command line syntax

The commands described in this book use the following special characters to define the syntax of commands:

- [] Identifies optional attributes. Attributes not enclosed in brackets are required.
- ... Indicates that you can specify multiple values for the previous attribute.
- | Indicates mutually exclusive information. You can use the attribute to the left of the separator or the attribute to its right. You cannot use both attributes in a single use of the command.
- { } Delimits a set of mutually exclusive attributes when one of the attributes is required. If the attributes are optional, they are enclosed in square brackets ([]).
- \ Indicates that the syntax in an example wraps to the next line.

The options for each command are listed alphabetically in the Options section, unless the options must be used in a specific order to implement the command.

wproxy

Works with endpoints in the endpoint proxy database and displays the version of the component installed.

Note: Never modify the attributes of an endpoint, reassign an endpoint to a new endpoint proxy, or remove an endpoint from the database when the endpoint proxy is running. These actions can corrupt the endpoint proxy database file (epproxy.bdb). Instead make a copy of the database and edit the copy:

1. Copy epproxy.bdb to an editable location.
2. Edit the copy using **wproxy -d /someDirectoryName edit**.
3. Stop the endpoint proxy (**epproxy.sh stop** or **net stop endpointproxy**)
4. Copy the edited file back into the endpoint proxy directory.
5. Start the endpoint proxy daemon/service (**epproxy.sh start** or **net start endpointproxy**)

Syntax

wproxy db [-d *db_directory*] edit *odnum* [*attribute=value*]

wproxy db [-d *db_directory*] ls [*odnum ...*]

wproxy db [-d *db_directory*] remove [*odnum ...*]

wproxy db edit *odnum* gwp_label=*label* proxy_port=*port*

wproxy db export *file_name* [*odnum ...*]

wproxy db import *file_name*

wproxy -v

Description

The **wproxy** command is run at the endpoint proxy. It lists endpoints, modifies attributes of an endpoint, moves endpoints from one endpoint proxy to another, and removes endpoints from the endpoint proxy database.

Before launching the **wproxy** command on UNIX operating systems, move to the directory where the endpoint proxy is installed and enter the following command:

```
. ./setup_env.sh
```

Options

-d *db_directory*

Indicates the directory where the database is located.

-v

Displays the version of the component that is installed on the machine from which it is run.

db

Indicates that you are performing actions on an endpoint proxy database.

edit

Edits an endpoint record in the endpoint proxy database.

export *file_name*

Exports endpoint data from the endpoint proxy database to the specified

file. You can specify one or more endpoints. If you do not specify an endpoint, all endpoint data is exported.

import *file_name*

Imports the endpoint data from the specified file into the endpoint proxy database.

ls Lists one or more endpoints in the endpoint proxy database.

remove

Removes one or more endpoints from the endpoint proxy database.

attribute=value

Indicates the attributes that you can modify, and the new value that you want to assign to the attribute. You can modify the following attributes:

proxy_port

Indicates the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. It is a port that the endpoint proxy uses to pose as the endpoint to the gateway.

real_port

Indicates the endpoint port that the endpoint uses to listen for connections from the gateway proxy. It is the same port as the `lcfcd_port` in the `last.cfg` file. Set this attribute using the **wproxy db edit** command if the `lcfcd_port` and the `real_port` get out of sync.

gwp_label

Indicates the label of the gateway proxy with which the endpoint connects.

odnum Indicates the number assigned to the endpoint by Tivoli Management Framework.

Authorization

Because the machine that the **wproxy** command runs on is probably not a machine with Tivoli Management Framework installed on it, authorization roles do not apply. On UNIX machines, run the **wproxy** command as root or the user account that was specified during the installation of the endpoint proxy. On Windows machines, log on as a member of the Administrators group or as TFSRsrvd to run the **wproxy** command.

Examples

To list the endpoints in the endpoint proxy database that is stored in the `/usr/epp` directory, enter the following command:

```
wproxy db -d /usr/epp ls
```

To edit a backup copy of `epproxy.bdb` and reassign the endpoint with an assigned endpoint number of 15763 to the gateway proxy named `x3gateway`, enter the following command:

```
wproxy -d /etc/Tivoli/epproxy/db-backup edit 15763 gwp_label=x3gateway
```

To make this change take effect, stop the endpoint proxy, replace the current `epproxy.bdb` with the edited version (in this example, the edited version is in `/etc/Tivoli/epproxy/db-backup`) and restart the endpoint proxy

This next example illustrates the use of the import and export options. For this example, assume the following:

- endpoints ep1, ep2, and ep3 log in through the gateway proxy named “mars”. The endpoint proxy for mars is “jupiter”. When the command, **wproxy db ls** is run on jupiter, its output will include lines that are similar to the following, for endpoints ep1, ep2, and ep3:

```
ep_label=ep1 odnum=88 address=1.2.3.4 real_port=9495 proxy_port=7003 gwp_label=mars
ep_label=ep2 odnum=92 address=1.2.3.5 real_port=9495 proxy_port=7010 gwp_label=mars
ep_label=ep3 odnum=97 address=1.2.3.6 real_port=9495 proxy_port=7015 gwp_label=mars
```

- An administrator wants ep1, ep2, and ep3 to log in through the gateway proxy named “pluto”. The endpoint proxy for pluto is “saturn”. When the command, **wproxy db ls** is run on saturn, the administrator notices that proxy_port 7003 and proxy_port 7015 are already assigned to endpoints impersonated by jupiter. To export the information for ep1, ep2, and ep3 into a file called jupiter-exports, the following command would be run on the endpoint proxy named jupiter:

```
wproxy db export jupiter-exports 88 92 97
```

Note: The exported information remains in jupiter’s database.

To import the information in the jupiter-exports file into epproxy.bdb on saturn, the following steps are required. Steps 2-7 should be executed on saturn.

1. Copy the export file, jupiter-exports, from jupiter to the directory on saturn where the endpoint proxy is installed.
2. Determine available ports that could be used as proxy ports for endpoints ep1 and ep3, since these endpoints specify proxy_port values that are already in use on saturn. The endpoint proxy uses the port-range value to assign proxy ports to the endpoints that it impersonates. The port-range value defaults to 7000-9000; a different range can be specified in epproxy.cfg. You can choose any available ports as proxy port values for ep1 and ep3; these ports can be within the port-range value, but this is not required. In this example, the administrator decides to assign ep1 a proxy_port value of 7101. ep3 is assigned a proxy_port value of 7255.
3. Edit the export file, jupiter-exports. Make the following changes:
 - for ep1, change the proxy_port value to 7101
 - for ep3, change the proxy_port value to 7255
 - for all three endpoints, change the gwp_label to pluto.
 Save the changes and close the jupiter-exports file.
4. The endpoint proxy on saturn should be stopped before modifications are made to it. Stop the endpoint proxy and make a backup copy of epproxy.bdb.
5. While the endpoint proxy is stopped, import the endpoint information in the jupiter-exports file by entering **wproxy db import jupiter-exports**.
6. View the contents of epproxy.bdb to verify that the endpoint information was imported. Enter **wproxy db ls**.
7. Restart the endpoint proxy so that the changes to epproxy.bdb take effect.

At this point, ep1, ep2, and ep3 are still impersonated by jupiter. Run the following commands to get the endpoints to log in through the pluto-saturn chain:

1. Set the interface list on each endpoint so that the new gateway proxy is listed first:

```
wep set interfaces -e ep1 pluto+nnnnn
wep set interfaces -e ep2 pluto+nnnnn
wep set interfaces -e ep3 pluto+nnnnn
```

where: *nnnn* is the gateway-port value used by pluto.

2. Point each endpoint at the new gateway proxy and restart the endpoint so it will use pluto as its gateway. To point the endpoints at pluto, use one of the following methods:

- Stop the endpoint. Edit the `last.cfg` file and add this line:

```
lcs.gateway_address=pluto+nnnn
```

Restart the endpoint.

- Specify the new `lcs.gateway_address` value when the endpoint is restarted:

```
wadminep ep1 reexec_lcf -D lcs.gateway_address=pluto+nnnn
```

where: *nnnn* is the gateway-port value used by pluto.

After the endpoints have restarted, you can confirm that they are logging in through the pluto gateway proxy by entering the following command:

```
wep ep1
```

The address value in the output should display saturn's IP address.

Appendix B. Troubleshooting

This appendix provides information about solving problems and gathering information to solve problems.

Testing proxy configuration

When the components are started, they try to exchange signals, called a *handshake*. The parent component sends its children a *who* request. The children reply. Similarly, the children send their parents a *tell* message and the parents reply. These exchanges enable the components in a chain of communication to establish the labels of all the components in the chain.

When one of the components is not running, the handshake fails. A message in the log file of each component lists the component with which the handshake failed.

Because components do not take the same amount of time to start, the log file might record a failure. The order of startup does not matter, but the order in which components are started affects the messages in the log file.

To test the components, set the log level to 3. The messages in the following example assume that you start the gateway proxy first. In the gateway proxy log file, look for lines of the type:

```
01/12/18 17:38:06 3 161 routingManager: WHO command received [l=null]
```

If the gateway proxy has problems connecting to its parent (relay or endpoint proxy), the log file records a message. For example, on Windows 2000 operating systems, the entry is similar to the following:

```
01/12/18 17:43:07 1 130 ERROR multiplex.newsessopen: cannot open connection (-1)
```

To verify that the gateway proxy handshake reached its parent, check the log file on the parent machine for an entry similar to the following:

```
01/12/18 17:53:34 3 248 routingManager: WHO reply command received [l=ascotti_gwp1]
```

Do a similar check with the parent (endpoint proxy or relay). Start the parent component first. When the components communicate, the log file shows entries for each child similar to the following (for endpoint proxies, this log file is epp.log):

```
01/12/18 17:51:27 3 47 routingManager: TELL command received [l=ascotti_gwp1]
```

If you do not see an entry similar to this, the components are not communicating, and the log file shows a message similar to the following:

```
01/12/18 17:43:07 1 130 ERROR multiplex.newsessopen: cannot open connection (-1)
```

To verify that the endpoint proxy or relay handshake reached its child, check the log file on the child machine for an entry similar to the following (for gateway proxies, this file is gwp.log):

```
01/12/18 17:40:16 3 179 routingManager: TELL reply command received
```

Debug each component individually to ensure that each is operating correctly. Do the following for each component:

1. Stop the component.

2. Restart the component you are testing and check the log file.

Debugging application errors

If you have a problem accessing an endpoint or running a management operation, verify that all the links in your communication path from the Tivoli server to the endpoint work correctly. Use the following checklist to diagnose problems:

- ___ 1. Ensure that the Tivoli server is running. The Tivoli server and the endpoint manager must be available for any transaction with an endpoint.
- ___ 2. Ensure that the gateways that manage endpoints in the DMZ are running. Use the **wgateway** command to verify this or use the endpoint manager user interface on the Tivoli desktop to check the gateway status. Check the gateway log (\$DBDIR/gatelog) on the gateway machine for messages indicating problems.
- ___ 3. Ensure that before starting your proxies, all machines that are involved in your deployment have DNS or IP visibility.
- ___ 4. Ensure that your firewall is correctly configured.
- ___ 5. Ensure that the endpoint proxy is configured to communicate with the correct gateway (address and port). When the endpoint proxy process starts, it logs a message stating the gateway IP and port with which it will communicate. See “Testing proxy configuration” on page 67 to ensure that the endpoint proxy is working correctly.
- ___ 6. Ensure that the gateway proxies and relays are configured to communicate with the correct endpoint proxy or relay. See “Testing proxy configuration” on page 67 to ensure proxy communication is working correctly.
- ___ 7. Ensure that the endpoints are running. Use the **wep** command to check the status of the endpoint. Check the `lcf.d.log` file on the endpoint for warnings and errors.

Using log files for troubleshooting

When you install the component, a log file is created in the directory in which you installed it:

epp.log

Logs messages for the endpoint proxy.

eventsink.log

Logs messages for the event sink.

gwp.log

Logs messages for the gateway proxy.

relay.log

Logs messages for the relay.

You can adjust the level of detail that you want reported in the logs. See Chapter 3, “Configuring the components,” on page 29 for instructions about setting the log level for each component.

In a production environment, use the default proxy logging level of 3. The range is 0–11. Values higher than 3 lower performance significantly.

Providing more detail in the log files

If you need to troubleshoot or to contact customer support, set the log levels of the components to a higher level of detail as follows:

- Set the gateway level to 7 by entering the following commands:


```
wgateway gateway_name set_debug_level 7
```

```
wgateway gateway_name restart
```
- Set the gateway proxy level to 8.
 - For UNIX operating systems:
 1. Edit the gwproxy.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 2. Enter the command: **./gwproxy.sh stop**
 3. Enter the command: **./gwproxy.sh start**
 - For Windows operating systems:
 1. Stop the gateway proxy.
 2. Edit the gwproxy.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 3. Start the gateway proxy.
- Set the endpoint proxy level to 8.
 - For UNIX operating systems:
 1. Edit the epproxy.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 2. Enter this command: **./epproxy.sh stop**
 3. Enter this command: **./epproxy.sh start**
 - For Windows operating systems:
 1. Stop the endpoint proxy.
 2. Edit the epproxy.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 3. Start the endpoint proxy.
- Set the event sink level to 8.
 - For UNIX operating systems:
 1. Edit the eventsink.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 2. Enter this command: **./eventsink.sh stop**
 3. Enter this command: **./eventsink.sh start**
 - For Windows operating systems:
 1. Stop the event sink.

2. Edit the eventsink.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 3. Start the event sink.
- Set the relay level to 8.
 - For UNIX operating systems:
 1. Edit the relay.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 2. Enter this command: **./relay.sh stop**
 3. Enter this command: **./relay.sh start**
 - For Windows operating systems:
 1. Stop the relay.
 2. Edit the relay.cfg file and change the debug-level entry to 8, and the **max-size** entry to 30. NOTE: Changing **max-size** to 30 will result in 60MB of disk space being consumed for the component log file and the component backup log file. Use a smaller number for **max-size** if disk space is an issue.
 3. Start the relay.
 - Set the endpoint level to 3

Using the Web interface edit Endpoint config to change **log_threshold** to 3.

Alternatively, edit the last.cfg file on the endpoint machine and change **log_threshold** to 3. Stop and restart the endpoint.

Interpreting the log files

The log files present information in the following format:

```
01/11/22 16:03:27 1 2144 ERROR tcpunidir.createServerSocket: \
cannot bind socket (10049)
```

The following table explains each column in the log file message:

Column	Description
1	Date
2	Time
3	Debug level of the message is logged
4	Thread ID
5	Message description

The debug level determines which messages are logged. You should debug problems that are logged at levels 0-3. Do not try to analyze messages at levels 5-11, because they are intended for customer support personnel. The following list defines the severity of each debug level:

- 0 Fatal Error. During or after startup, an application cannot continue.
- 1 Error. A single operation has failed.

- 2 Warning
- 3 Informational
- 4 Verbose. A trace of all the information exchanged between the endpoint and the gateway.
- 5 Light Debug. Shows function entries and exits.
- 6 This severity is currently not in use.
- 7 Debug
- 8 Communication Library
- 9 Intensive Communication Library
- 10 All Communication Library
- 11 Intensive Debug. This debug level should be used only when a deadlock issue is suspected. Do not use this debug level without first consulting with IBM support personnel.

Providing details to customer support

After you recreate the problem, provide the following information to IBM Customer Support:

- The error or exception message displayed and a description of the problem.
- The version of Tivoli Management Framework, applications, and patches installed. Use the **wlsinst** command. Note that the **wlsinst** command does not include information about where (or if) the Firewall Security Toolbox components are installed.
- A description of the configuration of all the Tivoli components installed.
- Details about the firewall and its configuration.
- Log files that you have gathered, including the log file from the endpoint manager (\$DBDIR/epmgrlog). See “Providing more detail in the log files” on page 68.
- The lcfld.log with debug level 3
- Also provide as many Firewall Security Toolbox component log files as possible. For example, if there are issues with an endpoint proxy, endpoint proxy log files are needed; log files from its children may provide very valuable information as well. Collect both .log and .log.bak files from the Firewall Security Toolbox components (for example, on an endpoint proxy, collect epp.log and epp.log.bak).
- Provide version information about the Firewall Security Toolbox components. This can be accomplished by running the following command and reporting the output of the command to support:

component -v

where *component* is **epproxy**, **gwproxy**, **relay**, or **eventsink**. For example, on a UNIX host, from the directory where a gateway proxy is installed, run **./gwproxy -v**

- The startup and configuration files of the components of the Tivoli Enterprise Firewall Security Toolbox:

epproxy.sh, epproxy.cfg

Script and configuration files for the endpoint proxy.

eventsink.sh, eventsink.cfg

Script and configuration files for the event sink.

gwproxy.sh, gwproxy.cfg

Script and configuration files for the gateway proxy.

relay.sh, relay.cfg

Script and configuration files for the relay.

- Optionally, if this indicates that there are errors: odstat output
- Optionally, if this indicates that there are errors: wtracelog output

Tuning

This section provides information about tuning Firewall Security Toolbox to increase the speed of distributions and to avoid distribution timeouts.

Tuning for large distributions

During software distributions, the endpoint proxy receives multiple packets from the gateway. The endpoint proxy sends the packets to the gateway proxy. After sending each packet that it receives from the gateway, the endpoint proxy waits for an acknowledgment from the gateway proxy. The endpoint proxy must wait for the acknowledgment from the gateway proxy before it can receive the next block of data from the gateway. You can increase the speed of large distributions by sending larger packets (and therefore fewer packets) between proxies, which decreases the number of times the endpoint proxy must wait for an acknowledgement.

To increase the speed of large distributions, follow these general steps:

1. Increase the MDist 2 **packet_size** parameter using the **wmdist** command.
2. Set the gateway-recv-buf-max parameter in the [endpoint-proxy] section of the endpoint proxy configuration file. This parameter allows the endpoint proxy to receive larger blocks of data from MDist 2 and therefore send larger blocks of data to its children. You should receive performance gains when the you set the **wmdist packet_size** to a value larger than its default value and set gateway-recv-buf-max to a value equal to or larger than **packet_size**.
3. Set the enable-tcp-nodelay configuration parameter to 1 in the [communication-layer] section of the configuration file for all proxies.
4. Set the tcp-buffer-size parameter in the [communication-layer] section of the configuration file for all proxies. Set this parameter to the same value as that of the gateway-recv-buf-max parameter that you set in step 2. This parameter specifies the buffer size that TCP/IP uses to send and receive data.

Tuning to reduce distribution timeouts

When complex configurations, numerous endpoints, or long response times cause your distributions to time out, you can change some of the timeout values to try to fix the problem. The following sections describe some timeout values that you can adjust to optimize the connections in your Tivoli environment.

Timeout values for Tivoli Management Framework

You can adjust the timeout values for Tivoli Management Framework to optimize the communication between the gateway and endpoints. For requests from the gateway to the endpoint, use the **wgateway** command to set the *session timeout*. This setting determines the amount of time (in seconds) that a gateway waits for a response from an endpoint after sending a request. The default is 300 seconds (5

minutes). Because responses from the endpoint might take longer when there are proxies between it and the gateway, a higher value would enable the gateway to wait longer for a response.

Timeout Values for the Firewall Security Toolbox

You can adjust timeout values to optimize the communication between the proxies. The **tcpip-timeout** value is the interval, in seconds, within which each component tries to complete a single operation with another component. The **tcpip-timeout** affects how long the endpoint proxy and gateway proxy wait to connect with their Tivoli Management Framework counterparts. For example, when the endpoint proxy connects to the gateway, it times out after the **tcp-timeout** interval is finished. Tune this parameter to give it time to connect to the gateway.

The **connect-timeout** value is the interval, in seconds, within which each component tries to connect to another component. Ensure that this value is not so low that the component does not have time to get a response from the other component. Tune this value to a value that is slightly longer than the longest a connection can take. For example, if it usually takes 10 seconds for the components to connect, set this value at 15 seconds. In unidirectional connections, the initiator sends the listeners a higher number of requests than in bidirectional connections, so the response time will be higher. However, if you make the value too high, the endpoint proxy takes longer to discover that the gateway proxy is down and to try a backup gateway proxy. For example, an endpoint proxy, which is an initiator in a unidirectional connection, has a gateway proxy a and 2 backup gateway proxies b and c and the **connect-timeout** is 30 seconds. If a and b are down, it will take 60 seconds (30 plus 30) for the endpoint proxy to try c.

For unidirectional configurations in which the child is the server and the parent is the client, configure the gateway proxy connection-timeout parameter to an appropriate value for the child to determine that the parent is unavailable. Set this parameter to a value that is larger than the value of the gateway proxy polling-interval parameter.

Connecting components from different versions

Firewall Security Toolbox, Version 1.3.2 can interoperate with Version 1.3.1. However, Firewall Security Toolbox, Version 1.2 cannot interoperate with Version 1.3.2 or Version 1.3.1. If you try to use incompatible versions of the Firewall Security Toolbox together, you get an error. Ensure that all the components in your configuration are compatible.

Rescuing lost endpoints from the gateway

Rescuing lost endpoints from the gateway Web page is not supported because the Web page does not go through the proxies.

Error when installing as user nobody

Problem: The following message is logged when you install a component on UNIX operating systems as user nobody and allocate a reserved port:

```
01/12/13 15:55:43 1 1 ERROR tcpbidir.createServerSocket: cannot bind socket (13)
01/12/13 15:55:43 0 1 FATAL tcpbidir.constructor: cannot create server socket
01/12/13 15:55:43 1 1 initRoutedSessionsManager: failure creating the connection
manager for child 0
01/12/13 15:55:43 0 1 routed sessions manager initialization failed
[cfg=eproxy.cfg;label=null]
```

Solution: Allocate port numbers that have permissions for the account being used to run the component.

Thread shortage on UNIX operating systems

Problem: On UNIX operating systems, if the number of threads that can be created simultaneously by a process is less than the number of threads needed by a particular Firewall Security Toolbox component, the component might fail to start correctly, stop running, or be unable to perform all its expected operations. For example, HP-UX systems have a default value of 64 for the **max_thread_proc** kernel parameter. This means that a process can run a maximum of 64 threads simultaneously.

The following message is logged when a Firewall Security Toolbox component runs out of threads:

```
runThread - pthread_create() failed (e=error_number) unable to create
thread, currently working threads number_of_threads
```

where:

error_number

Is the error number describing why the thread was not created.

number_of_threads

Indicates how many threads were running simultaneously when the thread failed to be created.

Solution: Increase the number of simultaneous threads allowed per process. The number of required threads depends on the component. The following formulas can help you determine the maximum number of threads needed to ensure that the Firewall Security Toolbox component works:

On the endpoint proxy, relay, or gateway proxy

$$\text{thread_number} = (\text{number_of_endpoints} \times 4 + 10) \times 1.25$$

where:

thread_number

Is the number of threads required by the component.

number_of_endpoints

The total number of endpoints that connect through the component. For example, if 500 endpoints connect through 2 gateway proxies to the same endpoint proxy, the number of endpoints value is 1000.

On the event sink

$$\text{thread_number} = (\text{max-sessions} + 10) \times 1.25$$

where:

thread_number

Is the number of threads required by the component.

max-sessions

Is the max-sessions value that you specify when you configure the event sink in the RECEPTION section.

These formulas ensure that the components work when they are operating under conditions of maximum workload. For example, if you there are 100 endpoints for

one endpoint proxy, the thread number indicates the maximum number of threads required if all the endpoints are sending downcalls and upcalls at the same time.

To understand how to increase the number of threads per process, see the documentation for the operating system installed on the machine.

Network Address Translation support

Dynamic NAT can be supported only on unidirectional clients where **enable-identity=1** is set in the component's configuration file. NAT is not supported for unidirectional servers or for components that communicate bidirectionally. See “Configuring enable-identity” on page 49 for more information.

Wake on LAN not supported

The Wake on LAN[®] feature is not supported with Firewall Security Toolbox

Gateway proxy label might be displayed incorrectly

Problem: In the output of the **wproxy** command, the gateway proxy label is displayed incorrectly when the endpoint proxy and gateway proxy machines use different locales, in particular if the machines use a double-byte character set (DBCS).

Solution: Avoid using DBCS characters in the gateway proxy label. Although this does not cause functional problems, it can cause the label to be displayed incorrectly in command line output.

Multicast feature not supported

The multicast feature of Tivoli Management Framework, Version 4.1 (or newer) is not supported with Firewall Security Toolbox.

Port conflicts

The range 7000–9000 is the default range of source ports that are used to initiate connections between the endpoint proxy and the gateway or between the gateway proxy and endpoints. It is recommended that you use ports outside of that range for communication between proxy components.

Gateway times out before distribution completed

Problem: When you have a complex configuration with multiple DMZs, your network can slow down significantly and some applications might take longer to distribute profiles.

Solution: To ensure that distribution takes place, increase the gateway timeout using the **wgateway set_session_timeout** command. See *Tivoli Management Framework Reference Manual* for the command usage.

Appendix C. Configuring ports for Firewall Security Toolbox

This section describes how Firewall Security Toolbox uses ports and provides information to help you configure these ports. This section also describes how to configure firewalls between Firewall Security Toolbox components.

Ports used by Firewall Security Toolbox

A TCP/IP connection between two peers requires two ports—a local (source) port and a remote (destination) port:

- A *source port* is the port that is allocated on the client side of a TCP/IP session.
- A *destination port* is the port that a client specifies to communicate with a remote peer over the network. This port is allocated on the server (listener) side of a TCP/IP session. This is the most important port for a firewall administrator. It impacts firewall configuration because Firewall Security Toolbox components must connect through a firewall to communicate with each other.

Except for the relay, each Firewall Security Toolbox component has a single destination port. The relay has two destination ports: one for its parent and one for its children. You configure the destination port for a parent component using the `children-remote-list` parameter in the configuration file for the child component. You configure the destination port for a child component using the `parent-remote-port` parameter in the configuration file for the child component.

You can configure Firewall Security Toolbox source ports using the `port-range` and `local-port-range` parameters. These parameters are located in the configuration file for each Firewall Security Toolbox component. The following sections describe these parameters and provide information about configuring them.

The local-port-range and port-range parameters

The `local-port-range` parameter specifies the range of source ports that are used to initiate communications between Firewall Security Toolbox components (endpoint proxies, gateway proxies, and relays). There is no default value for the `local-port-range` parameter. If `local-port-range` is not specified in the configuration file for the Firewall Security Toolbox component, a port is chosen at run time by the operating system. If `local-port-range` is specified in the component configuration file and none of the ports in that range are available when one proxy component attempts to connect to another proxy component, the connection attempt will fail. Messages similar to the following are logged:

```
03/10/03 11:52:44 1 324 tcpbidir.open: Address in use error. Cannot bind socket
432 to port 7777.
03/10/03 11:53:19 1 492 ERROR tcpbidir.open: cannot open connection
03/10/03 11:53:19 1 492 routedSessionCreate: [rs=10] open session failed (e=-1)"
```

The `port-range` parameter specifies the range of source ports that are used to initiate communications between a Firewall Security Toolbox endpoint proxy or gateway proxy and its Tivoli Management Framework counterpart (the gateway or endpoint, respectively). If `port-range` is not specified in the configuration file, the default range of 7000-9000 is used.

Note: Because relays connect only to other Firewall Security Toolbox components, they do not use the `port-range` parameter.

When specifying values for the local-port-range and port-range parameters, ensure that no ports are shared between the two parameters. For example, do not set local-port-range to 7000–7100 and port-range to 7050–8050, because ports 7050 through 7100 are shared between the two parameters. This can result in an insufficient number of available ports.

For endpoint proxies, the ports specified by the port-range parameter are used in two ways. In addition to specifying the range of source ports used to initiate communications with the gateway, the port-range parameter also specifies the listening ports that are allocated by the endpoint proxy for each endpoint it emulates. For example, if 100 endpoints log in through a particular endpoint proxy, 100 of the ports in port-range are used to listen for gateway connections that are directed towards a specific endpoint. When determining the number of ports to specify in the port-range for the endpoint proxy, you must account for source ports and listening ports.

Configuring source ports

When allocating port ranges, keep the following factors in mind:

- The TCP/IP TIME_WAIT state
- The Firewall Security Toolbox communications mode used (unidirectional or bidirectional)

When a connection is closed, the port associated with that connection is not always immediately available for reuse. This is because TCP/IP often puts a closed connection in TIME_WAIT state. Allocate extra ports in the port range to allow for the TIME_WAIT state to clear so that a source port will always be available.

You must also consider the communications mode employed, unidirectional or bidirectional, in port range allocation. Connections between Firewall Security Toolbox components are not persistent. For example, the connection between an endpoint proxy and a gateway proxy involved in a downcall to an endpoint is terminated after the downcall has completed. When bidirectional communication is used, either Firewall Security Toolbox component can initiate a connection. Bidirectional components typically initiate connections only after having been contacted by a Tivoli Management Framework counterpart; for example, a bidirectional gateway proxy would initiate a connection with its Firewall Security Toolbox parent after an endpoint initiates an upcall through the gateway proxy.

When unidirectional communication is used, only one Firewall Security Toolbox component is allowed to initiate communications with another Firewall Security Toolbox component. The component designated as the initiator connects with the listener component at regular intervals to determine whether data is available at the listener. (You set the interval at which the initiator connects with the listener component using the polling-interval parameter in the configuration file for the initiator.) Even when there is no communication activity between Firewall Security Toolbox and Tivoli Management Framework, the unidirectional initiator makes use of source ports to contact the listener component. This is not the case for bidirectional communications.

Observe the following guidelines when setting the local-port-range and port-range parameters for each Firewall Security Toolbox component:

Table 20.

Component	Configuration file section	Parameter	Guidelines
Endpoint proxy	endpoint-proxy	port-range	Use 2.5 to 3 ports per managed endpoint. This ratio allows for one listening port and one communication port for each managed endpoint, plus extra ports to account for TCP/IP TIME_WAIT situations. This value should be increased if a heavy load is expected. For example, if concurrent downcalls and upcalls are expected to most endpoints on a regular basis, use 4 ports per endpoint and specify 15 to 25% more ports to account for TIME_WAIT situations.
Endpoint proxy	children-cm-info	local-port-range	Use 2 to 2.5 ports per child Firewall Security Toolbox component. This value specifies the range of source ports used to open a connection between the endpoint proxy and its children (gateway proxies, relays, or both). When the endpoint proxy communicates as a unidirectional initiator, you might need to increase local-port-range, depending on the polling interval in use, to account for ports in TIME_WAIT state.
Gateway proxy	gateway-proxy	port-range	Use 1.5 to 2 ports per endpoint managed by the gateway proxy. This value specifies the range of source ports used to communicate with endpoints in response to downcalls from the gateway. If traffic between gateway and endpoints is high, increase this range.
Gateway proxy	parent-cm-info	local-port-range	Use 2 to 5 ports. This value specifies the range of source ports used to open a connection between the gateway proxy and its parent (a relay or endpoint proxy). You might need to increase this value if the gateway proxy communicates as a unidirectional initiator.
Relay	parent-cm-info	local-port-range	Use 2 to 5 ports. This value specifies the range of source ports used to open a connection between the relay and its parent (another relay or an endpoint proxy). You might need to increase this value if the relay communicates with its parent as a unidirectional initiator.
Relay	children-cm-info	local-port-range	Use 2 to 2.5 ports for each Firewall Security Toolbox child component (relays, gateway proxies, or both). You might need to increase this value if the relay communicates with its children as a unidirectional initiator, to account for ports in TIME_WAIT state.

Configuring firewalls

Firewalls between Firewall Security Toolbox components should be configured to accept communication between the following ports:

- Ports in the local-port-range of the parent component that connect to the listening port of each child component
- Ports in the local-port-range of the child component that connect to the listening port of the parent component

Appendix D. Understanding communication packets

This section describes Firewall Security Toolbox-specific packets that are sent between Firewall Security Toolbox peers (endpoint proxies, gateway proxies, and relays). It also describes how data received by these peers from Tivoli Management Framework counterparts is inspected and modified.

Understanding communication protocol

The peers communicate with each other using a proprietary protocol that allows multiplexing of multiple connections over a single session. Communications begin with an initial packet consisting of an ASCII string in the following format:

```
"%s %s %d %d %d %d %d %s\n"
```

In other words, the packet contains eight fields, delimited by a space characters (0x20) and ending with 0x0A. The fields are defined as follows:

1. *frame start token*, "<TFST_START>>"
2. *Firewall Security Toolbox version*, currently "1.3.2"
3. *frame command* (numeric), which currently takes on one of the following values:
 - 1 Indicates that the receiving peer should open a session
 - 2 Indicates that data is being sent
 - 3 Indicates that the receiving peer should close the session
 - 101 Indicates that the receiving peer successfully opened a session
 - 103 Indicates that the receiving peer successfully closed a session
4. *frame number* (numeric)
5. *session ID of the initiator*. This is a numeric value that is used in the multiplexing scheme.
6. *session ID of the listener*. This is a numeric value that is used in the multiplexing scheme.
7. *payload size*. If this value is non-zero, the payload buffer immediately follows the initial packet.
8. *extra data* (a string). This field currently takes on one of two values that indicates the session "type":
 - "RM" (for Routing Manager)—Indicates that the payload buffer following the initial packet is for either a **WHO** command or a **TELL** command. These commands are sent at peer startup time, before the peers are fully initialized.
 - "proxy"—Indicates that a peer is creating a routed session with another peer.

The initial packet is terminated by a newline character ('\n', 0x0A) and a trailing null character ('\0').

When the command value for the initial packet is 3, 101, or 103, payload size will be 0. When payload size field is non-zero, the buffer following the initial packet will have one of two formats:

- An ASCII string in the following format:

```
"%d %s"
```

In this case, the string contains the information about the **WHO** or **TELL** command or reply.

- The first field indicates the type of the string.
 - 1 Indicates a **WHO** command. Endpoint proxies and relays send **WHO** commands at startup time to each of their children.
 - 2 Indicates a **WHO** reply. A peer that receives a **WHO** command responds by sending a **WHO** reply to the sender of the **WHO** command.
 - 3 Indicates a **TELL** command. Gateway proxies and relays send a **TELL** command at startup time to their parent.
 - 4 Indicates a **TELL** reply. A peer that receives a **TELL** command responds by sending a **TELL** reply to the sender of the **TELL** command.
- The second field specifies one of two values:
 - The host name of the sender if the sender is a gateway proxy.
 - "." if the sender is an endpoint proxy or a relay.
- An ASCII string in the following format:
"%d %d %d %s %s %d %d"

In this case, the string contains information about the multiplexing scheme.

1. The first field (numeric) specifies the local session number.
2. The second field specifies the remote session number.
3. The third field specifies a command, as follows:
 - 1 Indicates that the remote peer should open a routed session.
 - 2 Indicates that synchronous data is being sent on the session.
 - 3 Indicates that the remote peer should close a routed session.
 - 4 Indicates a reply to another routed session command.
 - 5 Indicates that the remote peer should create a routed session.
 - 6 Indicates that a routed session should be destroyed.
 - 7 Indicates that asynchronous data is being sent on the session.
4. The fourth field specifies the host name of the local peer, or "." if the peer is not a gateway proxy.
5. The fifth field specifies the host name of the remote peer, or "." if the peer is not a gateway proxy.
6. The sixth field specifies the length of the payload buffer. If non-zero, the payload buffer immediately follows the seventh field in this string. This field typically has a value of 0 when the third (command) field has a value of 3 (close a routed session).
7. The seventh field specifies the sequence number of the packet.
8. The payload following the seventh field is typically data received from a gateway or endpoint that is being passed along appropriately (that is, up or down the proxy chain).

Inspecting Tivoli Management Framework packets

Firewall Security Toolbox components do a small amount of inspection or modification of the packets sent to them from their Tivoli Management Framework counterparts (such as, gateways and endpoints).

The gateway proxy inspects packets received from endpoints and rejects any that do not begin with the string "<START>".

The endpoint proxy inspects packets received from the gateway and rejects any that do not begin with the string "<START>" or "GET /" (such as, an HTTP packet).

The endpoint proxy maintains a database of the endpoints that it impersonates to the gateway. Among other things, this endpoint proxy database contains the IP address of each endpoint and the gateway proxy the endpoint is logging in through. The gateway sees all the impersonated endpoints as having the IP address of the endpoint proxy. The endpoint proxy accomplishes this by replacing the real IP address of the endpoint with its own IP address in packets sent to the gateway. Likewise, when the gateway sends information to the endpoint proxy that is destined for an endpoint impersonated by the endpoint proxy (such as, a downcall), references to the gateway are replaced by references to the appropriate gateway proxy before forwarding the information along to the endpoint.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/10111400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

Trademarks

IBM, Tivoli, Tivoli Enterprise, Tivoli Enterprise Console, TME, Wake on LAN, and AIX are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

accessibility vii

B

bidirectional connections

 considerations for configuring

 ports 78

 description 3

 installing UNIX endpoint proxy 11

 installing UNIX gateway proxy 12

 installing UNIX relay 13, 14

 installing Windows endpoint proxy 17

 installing Windows gateway proxy 19

 installing Windows relay 21, 23

bold typeface, meaning of viii

books

See publications

C

children-cm-info section

 configuring endpoint proxy 32

 configuring relay 41

command line syntax 61

commands

 TELL 81, 82

 WHO 81, 82

 winstlcf 57

 wproxy 62

communication

 between peers 81

 packets 81

 protocol 81

communication-layer section

 configuring endpoint proxy 31

 configuring gateway proxy 36

 configuring relay 40

compatibility between versions of

 proxies 25, 73

components

 child 5

 hierarchy 5

 multihomed hosts 9

 multiple instances 3

 parent 5

 prerequisite software 7

 uninstalling 26

 uninstalling relay, Windows

 systems 27

 uninstalling, UNIX systems 26

 uninstalling, Windows systems 26

 upgrading, UNIX systems 26

 upgrading, Windows systems 25

 where to install 8

configuration, proxy testing 67

configuring

 backup gateway proxy for endpoint proxy 47

 children-cm-info section on endpoint proxy 32

 children-cm-info section on relay 32, 41

 communication-layer section on

 endpoint proxy 31

 communication-layer section on

 gateway proxy 36

 communication-layer section on

 relay 40

 EIF section on event sink 45

 endpoint proxy 29

 endpoint proxy for backup gateway proxies 48

 endpoint-proxy section on endpoint proxy 29

 event sink 43

 event sink, for non-TME adapters 46

 gateway proxy 34

 gateway-proxy section on gateway proxy 34

 log section on endpoint proxy 30

 log section on event sink 45

 log section on gateway proxy 35

 log section on relay 39

 parent-cm-info section on gateway proxy 37

 parent-cm-info section on relay 42

 ports 77

 reception section on event sink 44

 relay 38

 relay section on relay 39

 sending section on event sink 44

customer support

See software support

Customer Support 71

D

data packets 81

DBCS characters in gateway proxy

 label 75

debugging application errors 68

E

EIF section, configuring event sink 45

endpoint

 configuring for backup gateway proxy 48

 in DMZ, connecting to gateway proxy 57

 installing in DMZ 57

 migrating to gateway proxy 47

 viewing properties by Web

 browser 59

 Web interface rescue 73

endpoint proxy

 backup gateway proxies, configuring 48

 backup gateway proxy, configuring 47

 children-cm-info section 32

 communication-layer section 31

 configuring 29

 description 1

 endpoint proxy port on UNIX systems 11

 endpoint proxy port on Windows systems 16

 endpoint-proxy section 29

 gateway port on UNIX systems 11

 gateway proxy host name on UNIX systems 11

 gateway proxy host name on Windows systems 16

 gateway proxy on Windows

 systems 16

 gateway proxy port on UNIX systems 11

 gateway proxy port on Windows systems 16

 inspecting packets 82

 installing on UNIX systems 10

 installing on Windows systems 15

 log section 30

 relay host name on UNIX systems 11

 relay host name on Windows systems 16

 relay port on UNIX systems 11

 relay port on Windows systems 16

endpoint proxy database

 aligning with endpoint manager

 database 57

 listing endpoints 54

 logging proxies 82

 modifying endpoint attributes 55

 removing endpoints 56

 working with endpoints 54

endpoint proxy host name

 on gateway proxy, UNIX systems 12

 on gateway proxy, Windows systems 19

 on relay, UNIX systems 13

 on relay, Windows systems 21

endpoint proxy port

 on endpoint proxy, UNIX systems 11

 on endpoint proxy, Windows systems 16

 on gateway proxy, UNIX systems 12

 on gateway proxy, Windows systems 19

epp.log 27, 68

epprox.ybdb 54, 56, 57

epprox.ycfg 27

event sink

 configuring 43

 configuring for non-TME adapters 46

- event sink *(continued)*
 - description 4
 - installing on UNIX systems 14
 - installing on Windows systems 24
 - LCF_DATDIR directory on UNIX systems 14
 - LCF_DATDIR directory on Windows systems 25
 - listening port on UNIX systems 14
 - listening port on Windows systems 25
- eventsink.cfg 27
- eventsink.log 27, 68

F

- failover to backup gateway proxies 47, 48
- files
 - epp.log 27
 - epproxy.cfg 27
 - eventsink.cfg 27
 - eventsink.log 27
 - gwp.log 27
 - gwproxy.cfg 27
 - relay.cfg 27
 - relay.log 27
 - uninstall.sh 26
- firewalls
 - bidirectional connections 3
 - configuring ports 80
 - in Tivoli environment 1
 - limited connections 3
 - relays in demilitarized zones 2
 - sending events 4
 - unidirectional connections 3

G

- gateway port
 - on endpoint proxy, UNIX systems 11
 - on endpoint proxy, Windows systems 16
- gateway proxy
 - configuring 34
 - configuring backup for endpoint proxy 47
 - configuring endpoint proxy for backup 48
 - description 1
 - endpoint proxy host name on UNIX systems 12
 - endpoint proxy host name on Windows systems 19
 - endpoint proxy port on UNIX systems 12
 - endpoint proxy port on Windows systems 19
 - failover 47
 - gateway proxy port on UNIX systems 12
 - gateway proxy port on Windows systems 19
 - inspecting packets 82
 - installing on UNIX systems 11
 - installing on Windows systems 17

- gateway proxy *(continued)*
 - label displayed incorrectly 75
 - relay host name on UNIX systems 12
 - relay host name on Windows systems 19
 - relay port on UNIX systems 12
 - relay port on Windows systems 19
- gateway proxy host name
 - on endpoint proxy, UNIX systems 11
 - on endpoint proxy, Windows systems 16
 - on UNIX relay 13
 - Windows child relay 22
- gateway proxy port
 - child on relay, UNIX systems 13
 - child on relay, Windows systems 22
 - on endpoint proxy, UNIX systems 11
 - on endpoint proxy, Windows systems 16
 - on gateway proxy, UNIX systems 12
 - on gateway proxy, Windows systems 19
- gateway timeout, troubleshooting 75
- gateway-proxy section, configuring
 - gateway proxy 34
- gwp.log 27, 68
- gwproxy.cfg 27

I

- initiator
 - installing endpoint proxy on UNIX systems 11
 - installing endpoint proxy on Windows systems 17
 - installing gateway proxy on UNIX systems 12
 - installing relay for child on UNIX systems 14
 - installing relay for child on Windows systems 23
 - installing relay for parent on UNIX systems 13
 - installing relay for parent on Windows systems 22
 - installing Windows gateway proxy 20
- installing
 - endpoint proxy, UNIX systems 10
 - endpoint proxy, Windows systems 15
 - event sink, UNIX systems 14
 - event sink, Windows systems 24
 - gateway proxy, UNIX systems 11
 - gateway proxy, Windows systems 17
 - on UNIX operating systems 10
 - on Windows systems 15
 - relay, UNIX systems 12
 - relay, Windows systems 20
 - TAR file 10
 - user account, endpoint proxy 11
 - user account, event sink 14
 - user account, gateway proxy 12
 - user account, relay 13
- introduction
 - simple Tivoli environment 1
 - Tivoli environment with demilitarized zones 2

- introduction *(continued)*
 - Tivoli environment with firewall 1
- italic typeface, meaning of viii

L

- LCF_DATDIR directory
 - installing event sink on UNIX systems 14
 - installing event sink on Windows systems 25
- listener
 - description 3
 - installing endpoint proxy on UNIX systems 11
 - installing endpoint proxy on Windows systems 17
 - installing gateway proxy on UNIX systems 12
 - installing relay for child on UNIX systems 14
 - installing relay for child on Windows systems 23
 - installing relay for parent on UNIX systems 13
 - installing relay for parent on Windows systems 22
 - installing Windows gateway proxy 20
- listening port
 - installing event sink on UNIX systems 14
 - installing event sink on Windows systems 25
- listing endpoints in endpoint proxy database 62
- local-port-range parameter
 - description 77
 - guidelines for configuring 78
- log files
 - debug levels 70
 - interpreting 70
 - level of detail 68
 - troubleshooting 68
- log section
 - configuring event sink 45
 - configuring gateway proxy 35
 - configuring relay 39

M

- manuals
 - See* publications
- migrating
 - endpoints in a DMZ to gateway proxy 57
 - endpoints to gateway proxy 47
 - to version 1.3.2 25
- modifying endpoints in endpoint proxy database 62
- monospace font, meaning of viii
- multicast support 75
- multihomed hosts 9

N

NAT support 75

O

online publications
 accessing vii
ordering publications vii
overview
 simple Tivoli environment 1
 Tivoli environment with demilitarized zones 2
 Tivoli environment with firewall 1

P

packets
 communication 81
 data 81
 inspecting 82
 structure 81
parent endpoint proxy port
 on relay, Windows systems 21
parent relay port
 on relay, Windows systems 21
parent remote port
 on relay, UNIX systems 13
parent-cm-info section
 configuring event sink 44
 configuring gateway proxy 37
 configuring relay 42
peer communication 81
port conflicts 75
port range, reserved 75
port-range parameter
 description 77
 guidelines for configuring 78
ports
 configuring 77
prerequisite software 7
 Tivoli Management Framework 7
proxy configuration, testing 67
publications
 accessing online vii
 ordering vii

R

reception section, configuring event sink 44
relay
 configuring 38
 description 2
 endpoint proxy host name on UNIX systems 13
 endpoint proxy host name on Windows systems 21
 gateway proxy host name (child) on Windows systems 22
 gateway proxy host name on UNIX systems 13
 gateway proxy port (child) on UNIX systems 13
 gateway proxy port (child) on Windows systems 22

relay (*continued*)
 installing on UNIX systems 12
 installing on Windows systems 20
 parent endpoint proxy port on Windows systems 21
 parent relay port on Windows systems 21
 parent remote port on UNIX systems 13
 relay host name (child) on Windows systems 22
 relay host name on UNIX systems 13
 relay host name on Windows systems 21
 relay port (child) on UNIX systems 13
 relay port (child) on Windows systems 22
 relay port for children machines on UNIX systems 13
 relay port for children machines on Windows systems 22
 relay port for parent machine on UNIX systems 13
 relay port for parent machine on Windows systems 21
relay host name
 child on relay, Windows systems 22
 on endpoint proxy, UNIX systems 11
 on endpoint proxy, Windows systems 16
 on gateway proxy, UNIX systems 12
 on gateway proxy, Windows systems 19
 on relay, UNIX systems 13
 on Windows relay 21
relay port
 child on relay, UNIX systems 13
 child on relay, Windows systems 22
 for children machines on relay, UNIX systems 13
 for children machines on relay, Windows systems 22
 for parent machine on relay, UNIX systems 13
 for parent machine on relay, Windows systems 21
 on endpoint proxy, UNIX systems 11
 on endpoint proxy, Windows systems 16
 on gateway proxy, UNIX systems 12
 on gateway proxy, Windows systems 19
relay section, configuring 39
relay.cfg 27
relay.log 27, 68
removing endpoints from endpoint proxy database 62
routing manager 81

S

sending events across firewalls 4
session types 81
software support
 contacting vii

starting components
 UNIX operating systems 53
 Windows operating systems 53
stopping components
 UNIX operating systems 53
 Windows operating systems 53
structure, packets 81
syntax
 command line 61

T

TCP/IP TIME_WAIT state 78
TELL command 81, 82
thread shortage error 74
timeout values
 for proxies, troubleshooting 73
 for Tivoli Management Framework, troubleshooting 72
Tivoli Distributed Monitoring, using with firewalls 5
Tivoli Enterprise Console Availability Intermediate Manager console 58
Tivoli Enterprise Console, using with firewalls 4
Tivoli software information center vii
troubleshooting
 application errors 68
 compatibility between versions 25, 73
 details for Customer Support 71
 gateway proxy label 75
 gateway timeout before distribution 75
 installing as user nobody 73
 log files
 debug levels 70
 interpreting 70
 levels 68
 list 68
 multicast support 75
 NAT support 75
 port conflicts 75
 thread shortage 74
 timeout values
 for proxies 73
 Tivoli Management Framework 72
 tuning 72
 tuning for large distributions 72
 tuning to reduce distribution timeouts 72
 Wake on LAN support 75
 Web interface for endpoints 73
typeface conventions viii

U

unidirectional connections
 considerations for configuring ports 78
 description 3
 installing UNIX endpoint proxy 11
 installing UNIX gateway proxy 12
 installing UNIX relay 13, 14

- unidirectional connections (*continued*)
 - installing Windows endpoint
 - proxy 17
 - installing Windows gateway
 - proxy 19
 - installing Windows relay 21, 23
- uninstall.sh file 26
- uninstalling relay, Windows systems 27
- uninstalling the components
 - backing up eproxy.bdb 26
 - UNIX systems 26
 - Windows systems 26
- UNIX systems
 - installing endpoint proxy 10
 - installing event sink 14
 - installing gateway proxy 11
 - installing on 10
 - installing relay 12
- upgrading the components
 - compatibility between versions 25
 - UNIX systems 26
 - Windows systems 25
- user account
 - endpoint proxy 11
 - event sink 14
 - gateway proxy 12
 - nobody, installing error 73
 - relay 13

V

- variables, notation for viii

W

- Wake on LAN support 75
- Web browser, viewing endpoint
 - properties 59
- Web interface, lost endpoints 73
- WHO command 81, 82
- Windows systems
 - installing endpoint proxy 15
 - installing event sink 24
 - installing gateway proxy 17
 - installing on 15
 - installing relay 20
- winstlcf command 57
- wproxy command 54, 55, 56, 62



Printed in USA

GC23-4826-02

