Tivoli® Management Framework

Firewall Security Toolbox Release Notes

Version 1.2

GI11-0901-00

Tivoli® Management Framework

Firewall Security Toolbox Release Notes

Version 1.2

GI11-0901-00

Tivoli Management Framework Firewall Security Toolbox Release Notes, Version 1.2

ISO 9001 Certification

This product was developed using an ISO 9001 certified quality system.

Certification has been awarded by Bureau Veritas Quality International (BVQI) (Certification No. BVQI - 92086 / A).

BVQI is a world leader in quality certification and is currently recognized by more than 20 accreditation bodies.

Contents

Preface

The Tivoli® Management Framework Firewall Security Toolbox provides a solution for managing your Tivoli network across firewalls without compromising security. The *Tivoli Management Framework Firewall Security Toolbox Release Notes* describe how to install and configure this feature of the Tivoli Management Framework.

Who Should Read These Release Notes

These release notes are for administrators and system programmers who configure the firewalls in their networks.

These release notes are 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 These Release Notes Contain

These release notes contain the following sections:

- "Introduction" on page xi

  Provides an overview of the main concepts.
- Chapter 1, "Installing the Tivoli Management Framework Firewall Security Toolbox" on page 1

  Provides instructions for the installing the components.
- Chapter 2, "Configuring the Components" on page 21

  Explains how to configure the components.
- Chapter 3, "Using the Tivoli Management Framework Firewall Security Toolbox" on page 35

  Explains how to perform various tasks, including starting and stopping the components.
- Appendix A, "Troubleshooting" on page 39

  Provides information to help identify and solve problems, including how to interpret the log files.

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 Maintenance and Troubleshooting Guide*, GC32-0394

  Explains how to maintain the Tivoli environment and troubleshoot problems that can arise during normal operations.

- *Tivoli Management Framework Planning for Deployment Guide*, GC32-0393

  Explains how to plan for deploying your Tivoli environment. It also describes Tivoli Management Framework and its services.

- *Tivoli Management Framework Reference Manual*, SC31-8434

  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 Tivoli-provided policy scripts.

- *Tivoli Management Framework Release Notes*, GI11-0836

  Provides late-breaking information about the Tivoli Management Framework.

- *Tivoli Management Framework User's Guide*, GC31-8433

  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.

Related Publications

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

- *Tivoli Enterprise Installation Guide*, GC32-0395

  Explains how to install and upgrade Tivoli Enterprise™ software within your Tivoli management 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 Enterprise Console Adapters Guide*, GC32-0668

  Provides detailed descriptions for the currently available Tivoli Enterprise Console® adapters.

Accessing Publications Online

You can access many Tivoli publications online at the Tivoli Customer Support Web site:

http://www.tivoli.com/support/documents/

These publications are available in PDF or HTML format, or both. Translated documents are also available for some products.

Ordering Publications

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

http://www.ibm.com/shop/publications/order

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

  http://www.tivoli.com/inside/store/lit_order.html

Providing Feedback about Publications

We are very interested in hearing about your experience with Tivoli products and documentation, and we welcome your suggestions for improvements. If you have comments or suggestions about our products and documentation, contact us in one of the following ways:

- Send an e-mail to pubs@tivoli.com.
- Complete our customer feedback survey at the following Web site:
  http://www.tivoli.com/support/survey/

Contacting Customer Support

If you have a problem with any Tivoli product, you can contact Tivoli Customer Support. See the *Tivoli Customer Support Handbook* at the following Web site:

http://www.tivoli.com/support/handbook/

The handbook provides information about how to contact Tivoli Customer Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting support

Conventions Used in These Release Notes

These release notes use several conventions for special terms and actions, operating system-dependent commands and paths.

Typeface Conventions

The following typeface conventions are used in these release notes:

Bold Lowercase and mixed-case commands, command options, and flags that appear within text appear like **this,** in **bold** type.

Graphical user interface elements (except for titles of windows and dialogs) and names of keys also appear like **this,** in **bold** type.

Italic Variables, values you must provide, new terms, and words and phrases that are emphasized appear like *this,* in *italic* type.

`Monospace` Commands, command options, and flags that appear on a separate line, code examples, output, and message text appear like `this`, in `monospace` type.

Names of files and directories, text strings you must type, when they appear within text, and HTML and XML tags also appear like `this`, in `monospace` type.

Operating System-dependent Variables and Paths

These release notes use 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.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Introduction

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

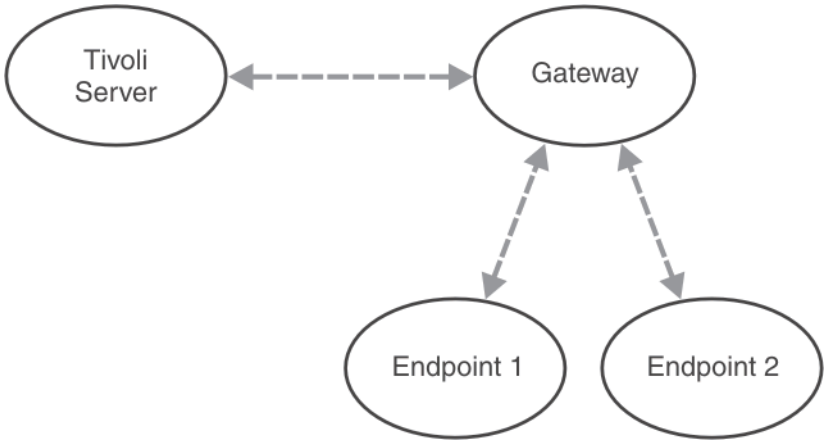


Figure 1. Tivoli Environment

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 gateway, the communication channels permitted by the firewall are limited. The Tivoli Management Framework Firewall Security Toolbox enables the endpoint and gateway to communicate across firewalls while respecting firewall restrictions.

Tivoli Environments with a Firewall

On the secure side of the firewall, the Tivoli Management Framework Firewall Security Toolbox provides an *endpoint proxy* that connects to the gateway as if it were the endpoint. On the less secure 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. The following Figure 2 on page xii shows a simple configuration with a single gateway proxy and endpoints.

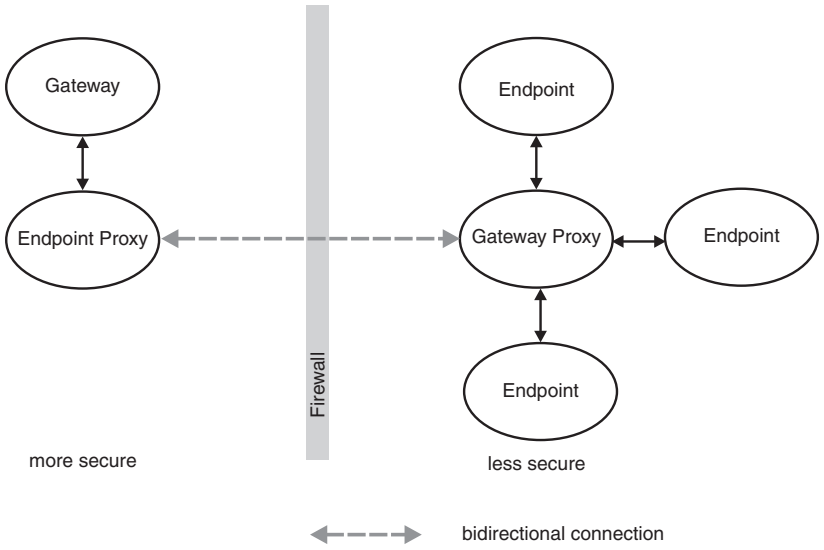


Figure 2. A Tivoli Environment with Endpoint Proxy and Gateway Proxy Connecting through a Single Firewall

Just as multiple endpoints can connect to a single gateway and multiple gateways to a single 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 components is based on a Tivoli proprietary protocol over TCP/IP.

Tivoli Environments 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, the Tivoli Management Framework 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 xiii shows an example.

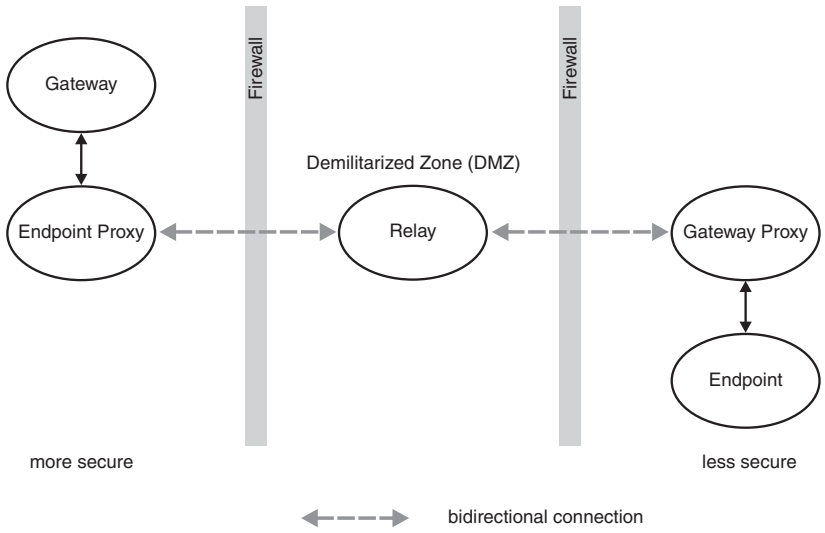


Figure 3. A Tivoli Environment with the Relay Connecting the Endpoint and Gateway Proxies through a DMZ

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 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 become the *listener*. This type of connection is known as *unidirectional*. The Tivoli Management Framework 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 uni- 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 xiv).

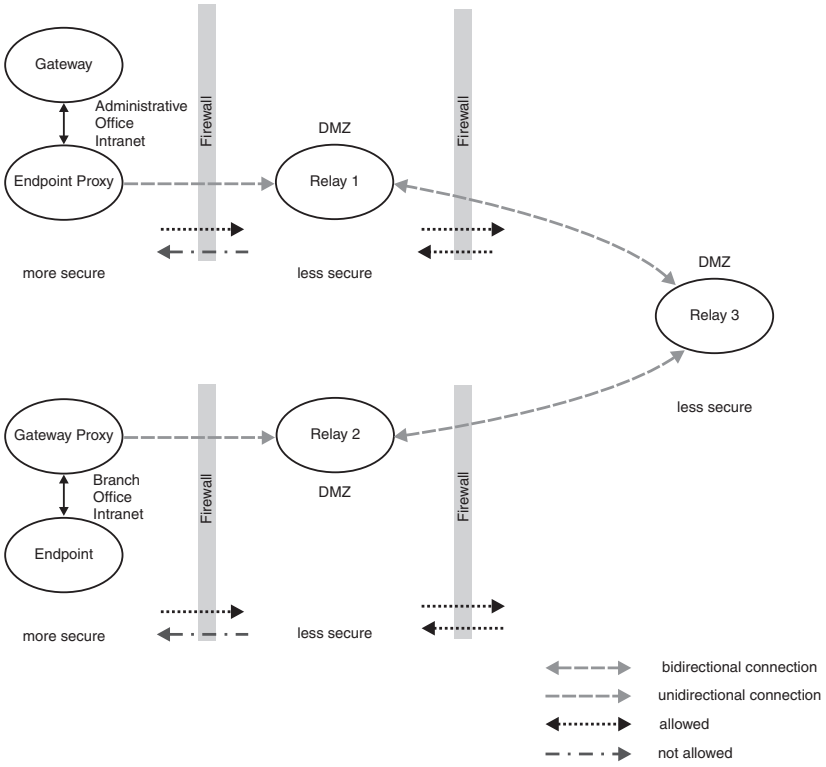


Figure 4. Example of an Environment with DMZs, Relays, and Bidirectional and Unidirectional Connections

From the more secure side of the branch office, the gateway proxy connects unidirectionally with the nearest relay and only the gateway proxy can initiate the connection. It is the *initiator* and the relay is the *listener*. The relays connect bidirectionally, and from the secure intranet of the administrative office, the endpoint proxy connects with its nearest relay. Similarly, the administrative office connects to the branch office.

Sending Events Across Firewalls

TME adapters use Tivoli endpoints to send events to the TEC server through Tivoli connections. When a firewall separates the endpoint from the gateway, 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 TEC server through non-Tivoli connections. When a firewall separates the non-TME adapter machine from the gateway, the Tivoli Management Framework Firewall Security Toolbox provides the *event sink*, which sends the events to the TEC server. The event sink, which is installed on an endpoint outside the firewall, collects events sent from non-TME adapters as if it were a TEC server and sends them to the TEC server as though they were TME events. The event sink can be connected by multiple non-TME adapters. See Figure 5 on page xv

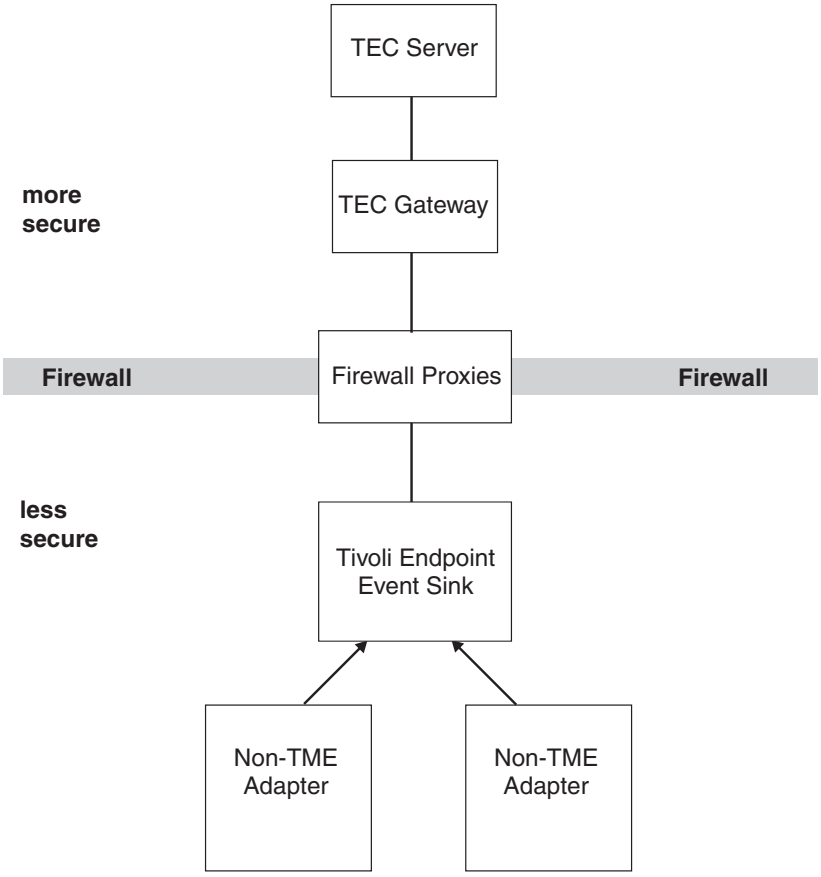


Figure 5. Event Sink Sends Non-TME Events to the TEC Server through the Firewall Proxies

In addition, the event sink can collect TME events that are sent from Tivoli Distributed Monitoring or the TEC Availability Intermediate Manager Console and forward them to TEC servers across firewalls.

Hierarchy of the Components

The hierarchy of the components of the Tivoli Management Framework Firewall Security Toolbox must be viewed in the context of the Tivoli region. The Tivoli server is the *parent* component. There can be only one server but multiple gateways and endpoints in a region. The endpoint proxy is the parent of the Tivoli Management Framework Firewall Security Toolbox components. It can have multiple *children*, relays and gateway proxies. The endpoint proxy connects into the Tivoli environment through the Tivoli gateway (see Figure 2 on page xii) and any communication that goes up the chain toward the endpoint proxy, and therefore toward the Tivoli gateway, goes toward the parent. For example, in Figure 3 on page xiii, 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 xiv), 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.

1

Installing the Tivoli Management Framework Firewall Security Toolbox

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

Prerequisite Software

Tivoli Management Framework Firewall Security Toolbox is a feature of the Tivoli Management Framework, Version 3.6.5 or 3.7.1. The components require the following operating system software to run:

Component	Supported Operating Systems
Endpoint proxy	■ Windows NT®, Version 4.0, with Service Pack 6 ■ Windows 2000 ■ AIX®, Version 4.3.3 ■ Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) ■ Sun Solaris, Version 2.7 or 2.8
Gateway proxy	■ Windows NT, Version 4.0, with Service Pack 6 ■ Windows 2000 ■ AIX, Version 4.3.3 ■ Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) ■ Sun Solaris, Version 2.7 or 2.8
Relay	■ Windows NT, Version 4.0, with Service Pack 6 ■ Windows 2000 ■ AIX, Version 4.3.3 ■ Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) ■ Sun Solaris, Version 2.7 or 2.8

Component	Supported Operating Systems
Event sink	■ Windows NT, Version 4.0, with Service Pack 6 ■ Windows 2000 ■ AIX, Version 4.3.3 ■ Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) ■ Sun Solaris, Version 2.7 or 2.8 ■ Tivoli Management Framework Endpoint 3.7.1 ■ Tivoli Enterprise Console 3.6.2 or 3.7

The event sink must be installed on a Tivoli endpoint. All other components of the Tivoli Management Framework Firewall Security Toolbox do *not* require any Tivoli software.

Planning Where to Install the Components

You can install as many gateways and endpoint proxies as you need in the firewall region. A recommended ratio is one Tivoli gateway and one endpoint proxy for every 500 endpoints in a region.

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 31 for configuration details.

Install a few endpoints first to test connectivity from the region to an endpoint through the proxy. Enter the following command from the 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 39.

Because the Tivoli Management Framework Firewall Security Toolbox requires a particular configuration of the region, it recommended that you keep machines that are in less secure zones in separate Tivoli regions. Set up a separate region to manage resources that are in a DMZ. To manage your endpoints as if they in a single 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 the Tivoli Management Framework 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 will work 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 Tivoli gateway that the endpoint proxy will use to communicate
  - The hostname or IP address of all the components you are installing
- Decide on some additional ports that the components will use to communicate with each other:
  - The endpoint proxy requires a connection port to receive traffic from the gateway proxies or relays. For more information, see "Configuring Backup Gateway Proxies" on page 31.
  - Gateway proxies require a port to receive traffic from the endpoint proxy or relay and one to listen for traffic from the endpoints.
  - Relays require a port to receive traffic from the components with which they connect, one for 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

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 Tivoli gateway in one subnet and relays or gateway proxies in another subnet (see Figure 6 on page 4).

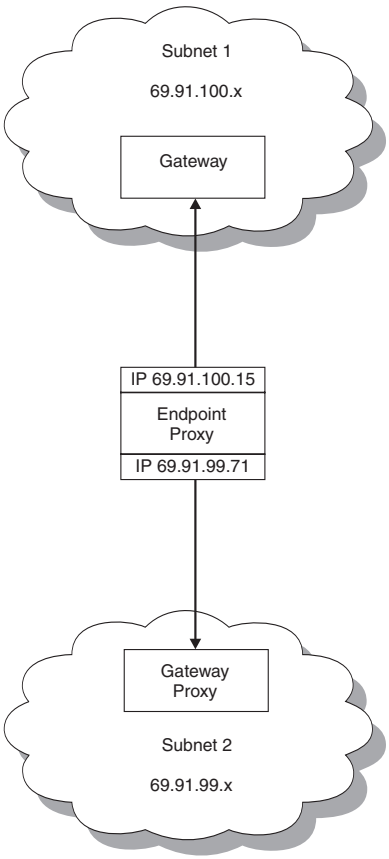


Figure 6. Example of a Multihomed Host

The endpoint proxy can connect to the Tivoli 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 2, “Configuring the Components” on page 21 for more details.

Getting the Installation Files

To install Tivoli Management Framework Firewall Security Toolbox, decompress the 1.2-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

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

Installing the Endpoint Proxy on UNIX

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/proxy]:`

      Specify the directory where the endpoint proxy should be installed. If the directory does not exist, you are asked whether it should be created.

   c. `User account which proxy should be run as [default=nobody]:`

      Specify the account name that will be used to run the proxy process. It is recommended that the account name selected be an unprivileged account.

   d. `Gateway address:`

      Specify the hostname or IP address of the Tivoli gateway with which the endpoint proxy communicates. The endpoint proxy can communicate with only one Tivoli gateway in a Tivoli management region.

   e. `Gateway port [default=9494]:`

      Specify the TCP/IP port number of the Tivoli gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. This is normally port 9494 and should not be changed unless the gateway is known to be using a different listening port with the Tivoli 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 hostname 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 hostname and port number of the additional destination machines. When you are done adding destination machines or 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.

   k. `Start the endpoint proxy? [Y/N] [default=Y]:`

      Enter `Y` to start the endpoint proxy. Otherwise, enter `N`.

Installing the Gateway Proxy on UNIX

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/proxy]:`

      Specify the directory where the gateway proxy should be installed. If the directory does not exist, you are asked whether it should be created.

   c. `User account which proxy should be run as [default=nobody]:`

      Specify a user name that will be used to run the proxy process. It is recommended that the account name selected be 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 Tivoli 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 hostname 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`.

Installing the Relay on UNIX

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/proxy]:`

      Specify the directory where the relay should be installed on the system. If the directory does not exist, you are asked whether it should be created.

   c. `User account which relay should be run as [default=nobody]:`

      Specify a user name that will be used to run the relay. It is recommended that the account name selected be an unprivileged account.

   d. `Relay port:`

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

   e. `Relay or endpoint proxy hostname:`

      Enter the hostname 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 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 parent machine only.

   h. `Relay port:`

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

   i. Relay or gateway proxy hostname:

      Specify the hostname 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 hostname and port number of the additional destination machines. When you are done adding destination machines or 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 parent 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 destination machine only.

m. `Start the relay? [Y/N] [default=Y]:`

Enter `Y` to start the relay. Otherwise, enter `N`.

Installing the Event Sink on UNIX

To install the event sink, follow these steps:

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 the event sink should be installed on the system. If the directory does not exist, you are asked whether it should be created.

   c. `LCF_DATDIR directory:`

   Specify the LCF_DATDIR directory of the endpoint on which you are installing the event sink.

   d. `User account which proxy should be run as [default=root]:`

   Specify a user name that will be used to run the event sink.

   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`.

Installing on Windows

The Tivoli Management Framework Firewall Security Toolbox provides a self-extracting EXE file to install each component on Windows 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 repairs.

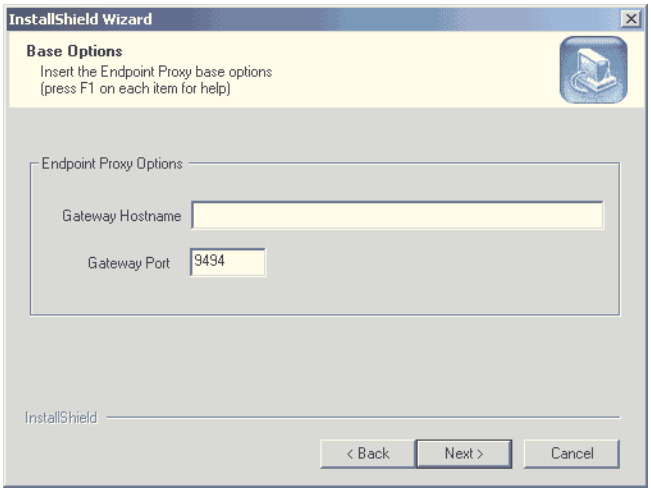
You can install in one of the following ways:

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

Installing the Endpoint Proxy on Windows

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, click **Yes** if you accept the license agreement.
4. On the next dialog, enter the installation directory and click **Next**. The dialog for Endpoint Proxy Options is displayed.



5. Complete the following fields:

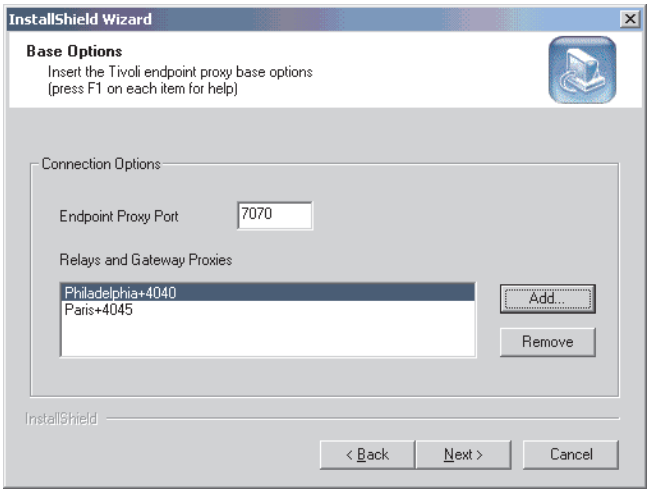
Gateway Hostname

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

Gateway Port

Enter the TCP/IP port number of the Tivoli 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 Tivoli endpoint.

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



6. Complete the following fields:

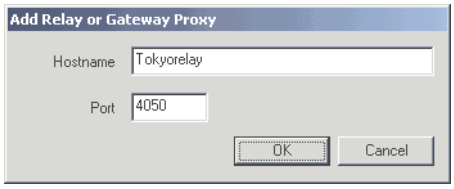
Endpoint Proxy Port

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**:

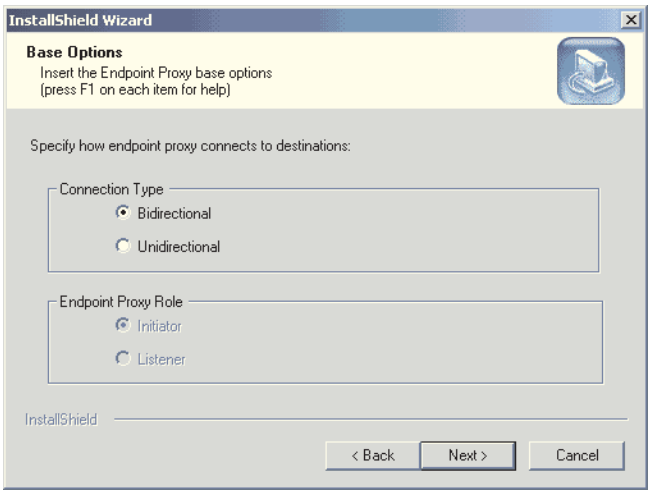
Hostname

Enter the hostname 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**.

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



8. Specify how the endpoint proxy connects to the destination relay or gateway 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 Endpoint Proxy Role box is enabled:

     Initiator
     The endpoint proxy machine can start the connection with the destination machine.

     Listener
     The destination machine can start the connection with the endpoint proxy machine.
9. Click **Next**. The next dialog shows a summary of your input.
10. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the endpoint proxy.
11. A message asks you whether or not you want to start the endpoint proxy. Click **Yes** to start it. Otherwise, click **No**.

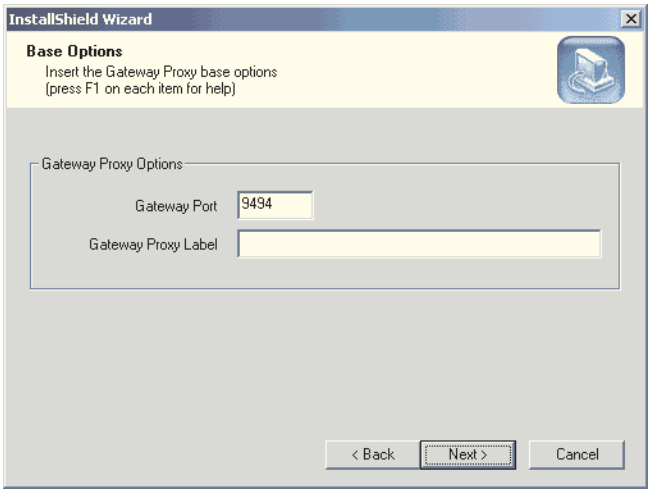
Installing the Gateway Proxy on Windows

The gateway proxy needs to be installed on a Windows NT or Windows 2000 host that is in the DMZ where the Tivoli endpoints will be located.

To install the gateway proxy, do the following:

1. From the directory that contains the Tivoli Gateway Proxy\w32-ix86\ subdirectory, double-click the Tivoli Gateway Proxy.exe file. The Tivoli Gateway Proxy InstallShield Wizard starts. Click **Next**.
2. On the next dialog, click **Yes** if you accept the license agreement.

3. On the next dialog, enter the installation directory and click **Next**. The dialog for Gateway Proxy Options is displayed.

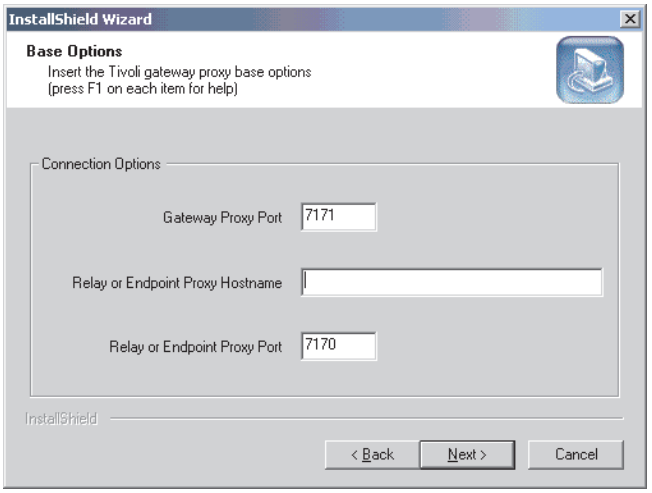


4. Complete the following fields:

   Gateway Port
   Enter the port number on the gateway proxy that represents the Tivoli 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.



5. Complete the following fields:

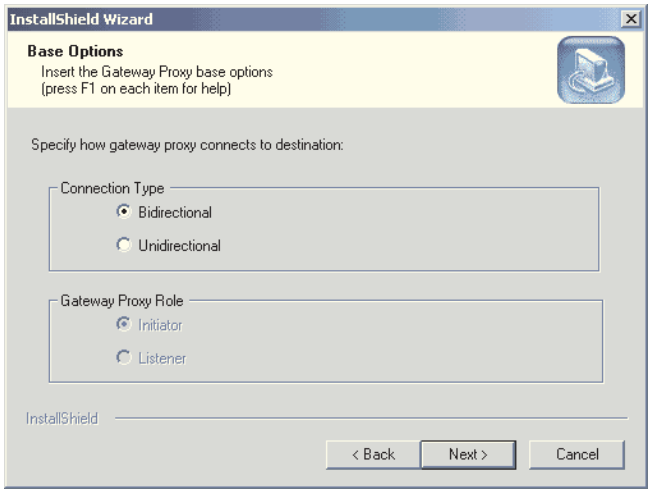
   Gateway Proxy Port
   Enter the port number that the gateway proxy uses to listen for connections from the relay or endpoint proxy.

   Relay or Endpoint Proxy Hostname
   Enter the hostname of the machine that the gateway proxy will connect to up the chain toward the Tivoli 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 Tivoli gateway.

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



6. 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

     The gateway proxy machine can start the connection with the parent endpoint proxy or relay machine.

     Listener

     The parent machine can start the connection with the gateway proxy machine.

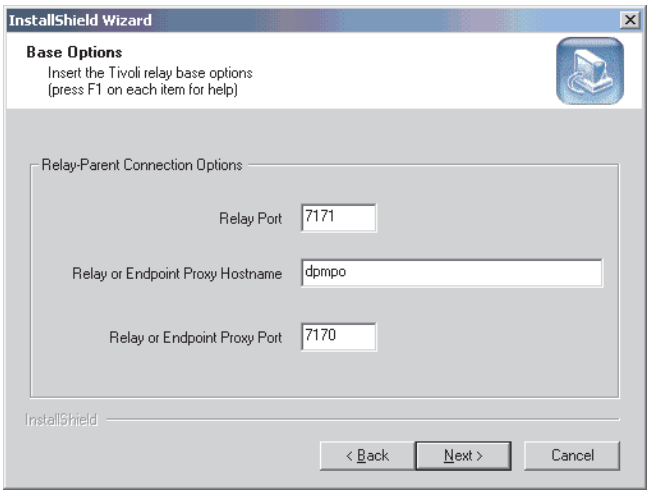
   Click **Next**. The next dialog shows a summary of your input.
7. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the gateway proxy.
8. A message asks you whether or not you want to start the gateway proxy. Click **Yes** to start it. Otherwise, click **No**.

Installing the Relay on Windows

To install the relay, do the following:

1. From the directory that contains the Tivoli Relay\w32-ix86\ subdirectory, double-click the Tivoli Relay.exe file. The Tivoli Relay InstallShield Wizard starts. Click **Next**.
2. On the next dialog, click **Yes** if you accept the license agreement.

3. On the next dialog, enter the installation directory and click **Next**. The dialog for Relay-Parent Connection Options is displayed.



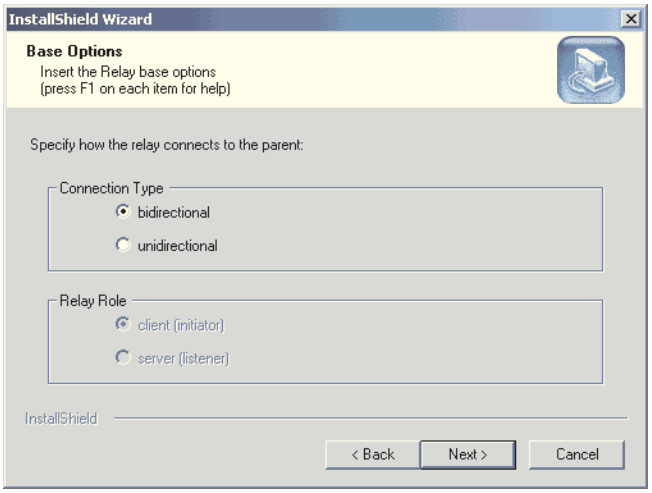
4. Complete the following fields regarding the connection between the relay and a parent machine, which can be either another relay or the endpoint proxy:

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

   Relay or Endpoint Proxy Hostname
   Enter the hostname 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.

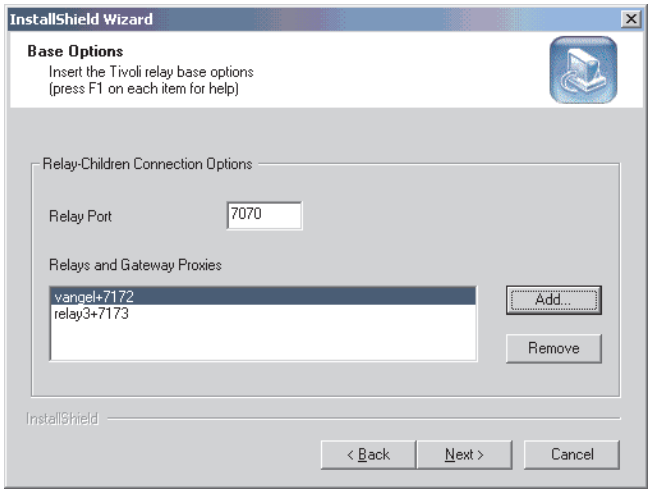


5. Specify how the relay connects to the parent relay or endpoint proxy:
   - Select **Bidirectional** to permit connections that are initiated by either machine.
   - Select **Unidirectional** to permit connections initiated by only one machine.
     If you select this option, the Relay Role box is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

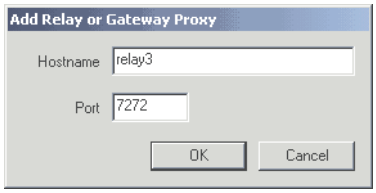


6. Complete the following fields:

   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.



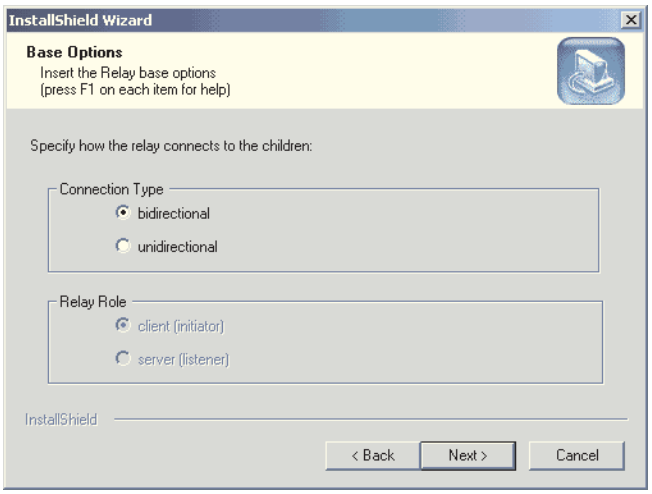
   Complete the fields and click **OK**:

   Hostname
   Enter the hostname 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.



7. Specify how the relay connects to the child relay or gateway 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 child machine.

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

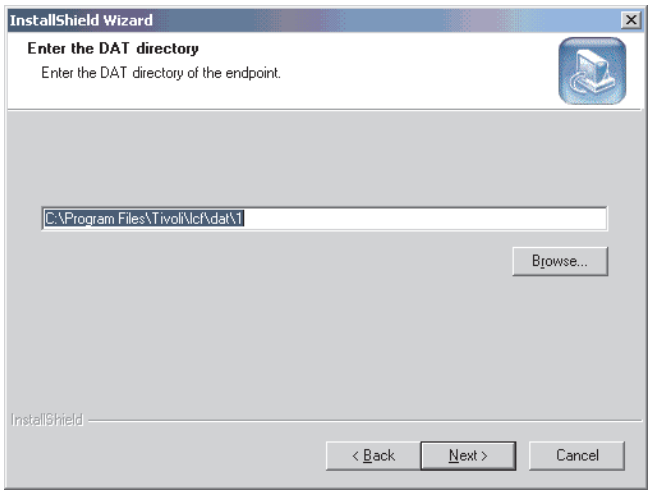
   Click **Next**. The next dialog shows a summary of your input.
8. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the relay.
9. A message asks you whether or not you want to start the relay. Click **Yes** to start it. Otherwise, click **No**.

Installing the Event Sink on Windows

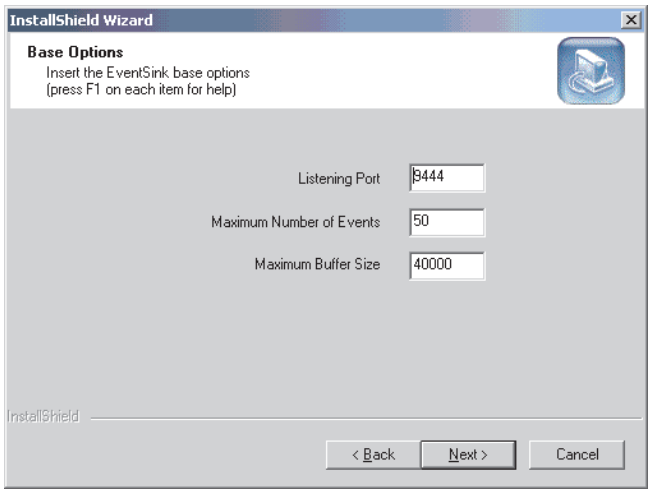
You must install the event sink on a Tivoli endpoint. To install the event sink, do the following:

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** if you accept the license agreement. The LCF_DATDIR dialog is displayed.



4. Specify the LCF_DATDIR directory of the endpoint on which you are installing the event sink. The event sink is installed in the lcf_datdir\..\..\bin\w32-ix86\mrt\ 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 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.

Uninstalling the Components

This section describes 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 `epproxy.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 `epproxy.bdb` file before uninstalling the endpoint proxy. After you reinstall the endpoint proxy, you can replace the `epproxy.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 36.

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

Uninstalling any component from UNIX requires the following tasks for all platforms:

1. Stopping the component. See "Starting and Stopping the Components" on page 35 for instructions.
2. Deleting the component directories and files.
3. Removing startup statements from system configuration files. The specific configuration files, directories, and symbolic links vary according to the platform.

Note: The directories specified in the following procedures assume the default locations.

Uninstalling from AIX

The following table lists the files and directories that you must remove for each component on AIX:

Files to Remove	Endpoint Proxy	Gateway Proxy	Relay	Event Sink
Installation directory	/usr/epproxy	/usr/gwproxy	/usr/relay	/usr/eventsink
Startup entry in /etc/inittab	Run the command: rmitab ″epproxy″	Run the command: rmitab ″gwproxy″	Run the command: rmitab ″relay″	Run the command: rmitab ″eventsink″
Files	/etc/rc.epproxy /etc/inittab.before.epproxy	/etc/rc.gwproxy /etc/inittab.before.gwproxy	/etc/rc.relay /etc/inittab.before.relay	/etc/rc.eventsink /etc/inittab.before.eventsink

Uninstalling from Linux

The following table lists the files and directories that you must remove for each component on Linux:

Files to Remove	Endpoint Proxy	Gateway Proxy	Relay	Event Sink
Installation directory	/usr/epproxy	/usr/gwproxy	/usr/relay	/usr/eventsink
Startup file	/etc/rc.d/init.d/epproxy	/etc/rc.d/init.d/gwproxy	/etc/rc.d/init.d/relay	/etc/rc.d/init.d/eventsink

Files to Remove	Endpoint Proxy	Gateway Proxy	Relay	Event Sink
Symbolic links	/etc/rc.d/rc2.d/S999epproxy /etc/rc.d/rc3.d/S999epproxy /etc/rc.d/rc4.d/S999epproxy /etc/rc.d/rc5.d/S999epproxy /etc/rc.d/rc0.d/K9epproxy /etc/rc.d/rc1.d/K9epproxy /etc/rc.d/rc6.d/K9epproxy	/etc/rc.d/rc2.d/S999gwproxy /etc/rc.d/rc3.d/S999gwproxy /etc/rc.d/rc4.d/S999gwproxy /etc/rc.d/rc5.d/S999gwproxy /etc/rc.d/rc0.d/K9gwproxy /etc/rc.d/rc1.d/K9gwproxy /etc/rc.d/rc6.d/K9gwproxy	/etc/rc.d/rc2.d/S999relay /etc/rc.d/rc3.d/S999relay /etc/rc.d/rc4.d/S999relay /etc/rc.d/rc5.d/S999relay /etc/rc.d/rc0.d/K9relay /etc/rc.d/rc1.d/K9relay /etc/rc.d/rc6.d/K9relay	/etc/rc.d/rc2.d/S99eventsink /etc/rc.d/rc3.d/S99eventsink /etc/rc.d/rc4.d/S99eventsink /etc/rc.d/rc5.d/S99eventsink /etc/rc.d/rc0.d/K9eventsink /etc/rc.d/rc1.d/K9eventsink /etc/rc.d/rc6.d/K9eventsink

Uninstalling from Solaris

The following table lists the files and directories that you must remove for each component on Solaris:

Files to Remove	Endpoint Proxy	Gateway Proxy	Relay	Event Sink
Installation directory	/usr/epproxy	/usr/gwproxy	/usr/relay	/usr/eventsink
Startup file	/etc/init.d/epproxy.rc	/etc/init.d/gwproxy.rc	/etc/init.d/relay.rc	/etc/init.d/eventsink.rc
Symbolic links	/etc/rc0.d/K49epproxy /etc/rc1.d/K49epproxy /etc/rc2.d/K49epproxy /etc/rc3.d/S999epproxy	/etc/rc0.d/K49gwproxy /etc/rc1.d/K49gwproxy /etc/rc2.d/K49gwproxy /etc/rc3.d/S999gwproxy	/etc/rc0.d/K49relay /etc/rc1.d/K49relay /etc/rc2.d/K49relay /etc/rc3.d/S999relay	/etc/rc0.d/K49eventsink /etc/rc1.d/K49eventsink /etc/rc2.d/K49eventsink /etc/rc3.d/S99eventsink

Uninstalling from Windows

You can uninstall the endpoint proxy or gateway proxy on a Windows NT or Windows 2000 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 **Start → Settings → 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**.

In addition, delete the CFG and LOG file from the directory in which the component is installed. The following table lists the name of each file for each component:

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

2

Configuring the Components

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

Configuring the Endpoint Proxy

After you install the endpoint proxy, the configuration file epproxy.cfg is created in the folder in which you installed the proxy. It contains the configuration input that you supply during installation. In addition, you can configure other options. To change these or configure other options, edit the epproxy.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

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

Keyword	Description
gateway-host	The address and port number of the Tivoli gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. This is normally port 9494 and should not be changed unless the gateway is known to be using a different listening port with the Tivoli endpoint. Use the format: *address+port_number*
gateway-interface	This option is used for multihomed endpoint proxies. It is the DNS or IP address of the endpoint proxies network interface that is used to communicate with Tivoli gateway network interface.
accept-timeout	Timeout interval, in seconds, that the endpoint proxy waits for connections initiated by the Tivoli gateway. Default: 300.
max-sessions	Number of connections that the endpoint proxy can manage at the same time. Default: 50.
tcpip-timeout	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. Default: 240.

Keyword	Description
port-range	Port ranges to use when allocating endpoint ports and when connecting with the gateway. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080` Default: 6000-8000.
database-path	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. Default: the directory where the component is installed.
disable-udp	Disables the endpoint proxy from forwarding UDP login requests to the Tivoli gateway by UDP. To *not* disable and to log in using TCP, specify 0. Default: 0.

Log

The [log] section lists log options. The following table lists the keywords and a description.

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 2 should only be used when recommended by Tivoli Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. Range: 0-11. Default: 3.
log-file	Full path of the log file where endpoint proxy messages are written. Installation default: epp.log. If no file specified, default: 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 *filename*.log.bak and a new log file is started. For no limit, specify 0. Default: 1.

Communication-layer

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

Keyword	Description
children-local-host	Network interface (DNS name or IP address) on endpoint proxy to listen for communication from its relays or gateway proxies.
children-local-port	Port number of the endpoint proxy machine from which it listens for connections with relays or gateway proxies.
children-remote-list	List of children hosts (relays or gateway proxies) to which the endpoint proxy connects. 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 endpoint proxy uses with its relays or gateway proxies. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.

Children-cm-info

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

Keyword	Description
connection-mode	Role of endpoint proxy in unidirectional connections only. Client is the initiator. Server is the listener. Values: client or server. Default: server.
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. Default: 17.
connect-timeout	Timeout, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. Default: 120.
log-mode	Logs buffer sent or received by the connection manager. Specifying a value other than 0 increases the number of messages that are logged significantly. This can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. Default: 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. Default: 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. Default: 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. Default: 5.

Configuring the Gateway Proxy

After you install the gateway proxy, the configuration file gwproxy.cfg is created in the folder in which you installed the proxy. It contains the configuration input that you supply during installation. In addition, you can configure other options. To change these or 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

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

Keyword	Description
gateway-port	Port number on the gateway proxy that represents the Tivoli gateway to the endpoints. Default: 9494.
gateway-interface	This option is used for multihomed gateway proxies. It is the DNS or IP address of the gateway proxies network interface used to communicate with the Tivoli endpoints.
tcpip-timeout	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. Default: 240.
proxy-label	Optional name to identify the gateway proxy instance. Default: *hostname*.
max-sessions	Number of connections that the gateway proxy can manage at the same time. Default: 50.
port-range	Port ranges to use when allocating ports to connect to endpoint ports. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080` Default: 6000-8000.

Log

The [log] section lists log options. The following table lists the keywords and a description.

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommend for normal operation. Levels higher than 2 should only be used when recommended by Tivoli Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. Range: 0-11. Default: 3.
log-file	Full path of the log file where gateway proxy messages are written. Installation default: gwp.log. If no file specified, default: 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 *filename*.log.bak and a new log file is started. For no limit, specify 0. Default: 1.

Communication-layer

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

Keyword	Description
parent-local-host	DNS name or IP address of gateway proxy from which it listens for connections from its relay or endpoint proxy.
parent-local-port	Port number of the gateway proxy machine from which it listens for connections with relay or endpoint proxy.
parent-remote-host	DNS name or IP address of the relay or endpoint proxy.

Keyword	Description
parent-remote-port	Port number of the parent relay or endpoint proxy from which it listens for connections with gateway proxy.
parent-cm-type	Communication interface and the connectivity that the gateway proxy uses with its relay or endpoint proxy. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.

Parent-cm-info

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

Keyword	Description
connection-mode	Role of gateway proxy in unidirectional connections only. Client is the initiator. Server is the listener. Values: client or server. Default: server.
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. Default: 17.
connect-timeout	Timeout, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. Default: 120.
log-mode	Logs buffer sent or received by the connection manager. Specifying a value other than 0 increases the number of messages that are logged significantly. This can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. Default: 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. Default: 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. Default: 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. Default: 5.

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 supply 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

The [relay] section is required at the top of the file, even though it contains no keywords.

Keyword	Description
tcpip-timeout	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. Default: 240

Log

The [log] section lists log options. The following table lists the keywords and a description.

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommend for normal operation. Levels higher than 2 should only be used when recommended by Tivoli Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. Range: 0-11. Default: 3.
log-file	Full path of the log file where relay messages are written. Installation default: relay.log. If no file specified, default: 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 *filename*.log.bak and a new log file is started. For no limit, specify 0. Default: 1.

Communication-layer

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

Keyword	Description
parent-local-host	DNS name or IP address of relay from which it listens for connections from its parent relay or endpoint proxy.
parent-local-port	Port number of the relay machine from which it listens for connections with parent relay or endpoint proxy.
parent-remote-host	DNS name or IP address of the parent relay or endpoint proxy.
parent-remote-port	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.

Keyword	Description
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. 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-cm-info

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

Keyword	Description
connection-mode	Role of relay in unidirectional connections only. Client is the initiator. Server is the listener. Values: client or server. Default: server.
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. Default: 17.
connect-timeout	Timeout, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. Default: 120.
log-mode	Logs buffer sent or received by the connection manager. Specifying a value other than 0 increases the number of messages that are logged significantly. This can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. Default: 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. Default: 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. Default: 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. Default: 5.

Parent-cm-info

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

Keyword	Description
connection-mode	Role of relay in unidirectional connections only. Client is the initiator. Server is the listener. Values: client or server. Default: server.
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. Default: 17.
connect-timeout	Timeout, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. Default: 120.
log-mode	Logs buffer sent or received by the connection manager. Specifying a value other than 0 increases the number of messages that are logged significantly. This can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. Default: 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. Default: 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. Default: 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. Default: 5.

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 adapter See “Configuring Non-TME Adapters for the Event Sink” on page 30. To configure AIM server see “Processing Events from the Tivoli Enterprise Console Availability Intermediate Manager Console” on page 37.

In addition, you can configure other options. To change these or 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

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

Keyword	Description
lcf-datdir	The `dat` directory of the Tivoli endpoint.
max-size-buffer	Maximum buffer size, in bytes, of the package that the event sink sends to the Tivoli Enterprise Console server. Default: 40000.
max-num-events-to-send	Enter the maximum number of events that the event sink sends to the Tivoli Enterprise Console server in a single package. Default: 50.
delay-time	Minimum interval, in seconds, between sending packages of events to the Tivoli Enterprise Console gateway. To send packages immediately, specify 0. Default: 1.
caching-timeout	Timeout, in seconds, by which the event sink sends events if neither the maximum buffer size nor the maximum number of events is reached. Default: 30.

RECEPTION

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

Keyword	Description
port	Port number on the endpoint where the event sink receives events. Default: 9444.
max-sessions	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. Default: 100.
tcpip-timeout	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 or the Tivoli Enterprise Console gateway. This timeout ensures that the event sink cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. Default: 240.
max-ram-cache	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 gateway. Default: 1024.
caching-timeout	Timeout, in seconds, by which the event sink sends events if neither the maximum buffer size nor the maximum number of events is reached. Default: 30.

EIF

The [EIF] section lists options for the Tivoli Enterprise Integration Facility. The following table lists the keywords and a description.

Keyword	Description
BufEvtMaxSize	The maximum size, in kilobytes, of the eventsink.cache file. Default: 64.

Keyword	Description
max-sessions	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. Default: 100.
BufEvtPath	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. Default: eventsink.cache.

LOG

The [log] section lists log options. The following table lists the keywords and a description.

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommend for normal operation. Levels higher than 2 should only be used when recommended by Tivoli Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. Range: 0-11. Default: 3.
log-file	Full path of the log file where event sink messages are written. Default for Windows: /*DAT_directory*/eventsink.log. Default for UNIX: /*installation_directory*/eventsink.log. If no file specified, default: standard error.
file-size	The maximum size, in megabytes, that the log file can reach. For no limit, specify 0. Default: 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=`*`hostname`*

`ServerPort=`*`port`*

Where:

hostname
: The hostname of the endpoint on which the event sink is installed

port
: The port on which the event sink listens for events.

Migrating Endpoints from a Tivoli Gateway 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 hostnamegwp+port
     ```

     Where:

     ep_label
     Is the label of the endpoint

     hostnamegwp
     Is the hostname 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, enter the command line options for lcfd to set the gateway –g and/or login interfaces –Dlcs.login_interfaces. Note that you must know the HTTP user name and password for the endpoint. See the *TME Planning and Installation 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 are *not* supported for endpoints and gateways that have the firewall proxies between them:

- The **wep migrate** command
- The **select_gateway_policy** command

These two framework features cannot be used during the regular process of the proxies.

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.
3. 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.
4. 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`.

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.

To the login policy, add the following:

- The logic that produces the list of gateway proxies for the endpoint, for which the login policy is called, in the following format:

  gateway_proxy1+port1:gateway_proxy2+port2

  gateway_proxy1+port1
  : Indicates the hostname or IP address of the main gateway proxy and its port number.

  gateway_proxy2+port2
  : Indicates the hostname 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
  wep sync_gateways
  ```

  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.

For new endpoints: When you first install an endpoint, specify the gateway proxie 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_lcfd
```

Where *endpoint_label* is the label of the endpoint.

Configuring the Tivoli Environment

This section describes other parameters in the Tivoli environment that must be configured for the Tivoli Management Framework Firewall Security Toolbox.

Setting the Endpoint Proxy Login Interval on All Platforms

The Tivoli endpoint manager prevents multiple logins from the same IP address within a specified interval. The Tivoli gateways in the region share the interval. Using a single IP address and multiple ports, the endpoint proxy represents each endpoint to the Tivoli gateway. To allow endpoints (and therefore the endpoint proxy) to login with the same IP address, set the login interval value to 0 seconds in the Tivoli endpoint manager configuration.

You need to have authorization to run this command and you need to set up the environment by sourcing the `setup_env.sh` file from a Bourne shell prompt (`$BINDIR/tools/bash.exe` on Windows NT).

For Tivoli Management Framework 3.6.5, do the following:

1. Type:

   ```
   epmgr='wlookup EndpointManager'
   idlcall $epmgr _set_login_interval 0
   ```

2. Type:

   ```
   odadmin reexec all
   ```

To restart the region.

For Tivoli Management Framework 3.7, do the following:

1. Type:

   ```
   wepmgr set login_interval=0
   ```

2. Type:

   ```
   odadmin reexec all
   ```

To restart the region.

3

Using the Tivoli Management Framework Firewall Security Toolbox

This chapter describes how to work with the Tivoli Management Framework 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 the Tivoli Management Framework Firewall Security Toolbox.

On **Windows** 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
3. Select **Start** from the pop-up menu.

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

On **UNIX** 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

To stop the components on UNIX, enter the command:

`./component.sh stop`

Working with Endpoints Logged in through the Proxy

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 command `wproxy`. 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 command:

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

Where `db_directory` indicates the directory in which the database is stored.

The results appear in the following format:

```
odnum=identifier address=IP_address proxy_port=port1 real_port=port2
```

where:

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

IP_address
: Indicates the address of the endpoint.

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 Tivoli gateway.

port2
: Indicates the endpoint port that the gateway proxy uses to communicate with the endpoint.

Removing an Endpoint from the Database

When endpoints are deleted from a 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 the Tivoli Management Framework.

Backing Up and Restoring the Endpoint Manager Database

When you back up the Endpoint Manager database using the wbkupdb command, the endpoint proxy database of endpoints 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

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. Recommendations for alternative methods follow:

- 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 rexec port on UNIX endpoints.
- Create a separate Tivoli region in the DMZ for installing endpoints only. Shut down the region after installation and leave it dormant unless you need to install other endpoints. Then, configure the endpoints to communicate with the gateway proxy.

To migrate endpoints from a Tivoli gateway to a gateway proxy, do the following:

1. Delete the endpoint from the region to which it is connected using the wdelep command.
2. Stop the endpoint.
3. From the endpoint DAT directory, delete all *except* the following files:
   - last.cfg
   - lcf_env.csh
   - lcf_env.sh
   - lcfd.sh
4. Edit the last.cfg file:
   a. 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.
   b. Add the following entry:
      `lcs.login_interfaces=`*`gateway_proxy_hostname+port`*
      Where *gateway_proxy_hostname+port* is the hostname and the port number of the gateway proxy.
5. 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 the Tivoli Enterprise Console Availability Intermediate Manager Console installed, you need to configure it to work with the Tivoli Management Framework Firewall Security Toolbox. When there is a firewall between the machine with the Tivoli Enterprise Console Availability Intermediate Manager Console 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 the 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 hostname 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 hostname 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 hostname 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 hostname and port number of the endpoint in the Location text box.

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

`http://hostname:port_number`

Where:

hostname
: Indicates the hostname 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.

A

Troubleshooting

This chapter 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 NT or Windows 2000, 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 show entries for each child similar to the following:

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

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:

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

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 diagnose problems:

__ 1. Ensure that the server is running. The 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/IP visibility.

__ 4. Ensure that your Firewall is correctly configured.

__ 5. Ensure that the endpoint proxy is configured to communicate with the correct Tivoli 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 39 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 39 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 `lcfd.log` file on the endpoint for warnings and errors.

Using the 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 2, “Configuring the Components” on page 21 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

However, if you need to troubleshoot or to contact Tivoli 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:
    1. Edit the gwp.cfg file and change the debug-level entry to 8.
    2. Enter the command: **./gwproxy.sh stop**
    3. Enter the command: **./gwproxy.sh start**
  - For Windows:
    1. Stop the gateway proxy.
    2. Edit the gwproxy.cfg file and change the debug-level entry to 8.
    3. Start the gateway proxy.
- Set the endpoint proxy level to 8.
  - For UNIX:
    - Edit the epp.cfg file and change the debug-level entry to 8.
    - Enter this command: **./epproxy.sh stop**
    - Enter this command: **./epproxy.sh start**
  - For Windows:
    1. Stop the endpoint proxy.
    2. Edit the epproxy.cfg file and change the debug-level entry to 8.
    3. Start the endpoint proxy.
- Set the event sink level to 8.
  - For UNIX:
    - Edit the eventsink.cfg file and change the debug-level entry to 8.
    - Enter this command: **./eventsink.sh stop**
    - Enter this command: **./eventsink.sh start**
  - For Windows:
    1. Stop the event sink.
    2. Edit the eventsink.cfg file and change the debug-level entry to 8.
    3. Start the event sink.
- Set the relay level to 8.
  - For UNIX:

- Edit the relay.cfg file and change the debug-level entry to 8.
- Enter this command: **./relay.sh stop**
- Enter this command: **./relay.sh start**

• For Windows:

1. Stop the relay.
2. Edit the relay.cfg file and change the debug-level entry to 8.
3. Start the relay.

■ lcfd level 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 Tivoli Customer Support experts. 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

Providing Details to Tivoli Customer Support

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

- The error or exception message displayed and a description of the problem.
- The version of the Tivoli Management Framework, applications, and patches installed. Use the **wlsinst** command.
- 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 41.
- The startup and configuration files of the components of the Tivoli Management Framework 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: `wtrace output`

Tuning

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 the Tivoli Management Framework

You can adjust the timeout values for the 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 Tivoli Management Framework 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 Tivoli 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 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`.

Rescuing Lost Endpoints from the Gateway

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

Error on UNIX Systems When Installing as User Nobody

Problem: The following message is logged when you install a component on UNIX 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=epproxy.cfg;label=null]
```

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

NAT Not Supported

The network address translation (NAT) feature does not work with the Tivoli Management Framework Firewall Security Toolbox.

Wake on LAN Not Supported

The Wake on LAN feature is not supported with the Tivoli Management Framework Firewall Security Toolbox.

Tivoli Gateway Times Out before Distribution Complete

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 Tivoli gateway timeout using the wgateway set_session_timeout command. See the *Tivoli Management Framework*

Reference Manual for the command usage.

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GI11-0901-00