CHAPTER 4

Using Syslog

This chapter presents an overview of the syslog protocol and shows you how to deploy an end-to-end syslog system. The chapter includes a discussion about the syslog architecture and discusses deploying syslog servers in Linux and Windows OSs with a focus on their relevance in a Cisco environment. Also included are the steps involved in configuring Cisco devices for syslog.

Overview of Syslog

The syslog protocol, defined in RFC 3164, was originally written by Eric Allman. This protocol provides a transport to allow a device to send event notification messages across IP networks to event message collectors, also known as syslog servers. The protocol is simply designed to transport these event messages from the generating device to the collector. The collector doesn't send back an acknowledgment of the receipt of the messages.

In a UNIX operating system, the kernel and other internal components generate messages and alerts. These messages are typically stored in a file system or relayed to another device in the form of syslog messages. The internal daemon, called Syslogd, handles the syslog process. This daemon is an integral part of most UNIX/Linux distributions and does not need to be downloaded or installed. Syslog provides a central point for collecting and processing system logs. These system logs are useful later for troubleshooting and auditing. For example, when a hacker breaks into a system, the trail left behind by the hacker's activity is logged in the syslog messages. These messages can then be used to understand the attack, assess the damage, and patch the system.

Various Cisco devices, including routers, switches, PIX Firewalls, VPN concentrators, and so on, generate syslog messages for system information and alerts. For example, a Cisco router can generate a syslog message when an interface goes down or the configuration is changed. Similarly, a Cisco PIX Firewall can generate a syslog message when it blocks a TCP connection. Cisco devices can be configured to send the syslog messages to an external machine that acts as a central syslog server. However, if the connectivity between the Cisco device and the syslog server is down, no syslog messages would be captured by the server. In such cases, the syslog messages stored locally by the Cisco devices are the only source of information to determine the root cause of the issue.

Syslog uses the User Datagram Protocol (UDP), port 514, for communication. Being a connectionless protocol, UDP does not provide acknowledgments. Additionally, at the application layer, syslog servers do not send acknowledgments back to the sender for receipt of syslog messages. Consequently, the sending device generates syslog messages without knowing whether the syslog server has received the messages. In fact, the sending devices send messages even if the syslog server does not exist.

The syslog packet size is limited to 1024 bytes and carries the following information:

- Facility
- Severity
- Hostname
- Timestamp
- Message

A clear understanding of each of the syslog packet parameters can help you easily deploy syslog systems across your network. Note that the first two parameters, facility and severity, are often misunderstood.

Facility

Syslog messages are broadly categorized on the basis of the sources that generate them. These sources can be the operating system, the process, or an application. These categories, called facility, are represented by integers, as shown in Table 4-1. The local use facilities are not reserved and are available for general use. Hence, the processes and applications that do not have pre-assigned facility values can choose any of the eight local use facilities. As such, Cisco devices use one of the local use facilities for sending syslog messages.

Table 4-1 *Facility Values*

Integer	Facility
0	Kernel messages
1	User-level messages
2	Mail system
3	System daemons
4	Security/authorization messages
5	Messages generated internally by Syslogd
6	Line printer subsystem
7	Network news subsystem
8	UUCP subsystem
9	Clock daemon

Table 4-1 *Facility Values (Continued)*

Integer	Facility
10	Security/authorization messages
11	FTP daemon
12	NTP subsystem
13	Log audit
14	Log alert
15	Clock daemon
16	Local use 0 **(local0)**
17	Local use 1 **(local1)**
18	Local use 2 **(local2)**
19	Local use 3 **(local3)**
20	Local use 4 **(local4)**
21	Local use 5 **(local5)**
22	Local use 6 **(local6)**
23	Local use 7 **(local7)**

By default, Cisco IOS devices, CatOS switches, and VPN 3000 Concentrators use facility local7 while Cisco PIX Firewalls use local4 to send syslog messages. Moreover, most Cisco devices provide options to change the facility level from their default value.

Severity

The source or facility that generates the syslog message also specifies the severity of the message using a single-digit integer, as shown in Table 4-2.

Table 4-2 *Severity Values*

Integer	Severity
0	Emergency: System is unusable.
1	Alert: Action must be taken immediately.
2	Critical: Critical conditions.
3	Error: Error conditions.
4	Warning: Warning conditions.
5	Notice: Normal but significant condition.
6	Informational: Informational messages.
7	Debug: Debug-level messages.

Cisco devices use severity levels of Emergency to Warning to report software or hardware issues. A system restart or interface up/down messages are sent through the Notice level. A system reload is reported through the Informational level. The output of debug commands is expressed through the Debug level.

Hostname

The hostname field consists of the host name (as configured on the host itself) or the IP address. In devices such as routers or firewalls, which use multiple interfaces, syslog uses the IP address of the interface from which the message is transmitted.

Timestamp

The timestamp is the local time, in MMM DD HH:MM:SS format, of the device when the message was generated. Although RFC 3164 does not specify the use of a time zone, Cisco IOS allows configuring the devices to send the time-zone information in the message part of the syslog packet. Such timestamps are generally prefixed with a special character, such as an asterisk (*) or colon (:), to prevent the syslog server from misinterpreting the message. The timestamp format, including the time-zone information, is MMM DD HH:MM:SS Timezone *.

NOTE For the timestamp information to be accurate, it is good administrative practice to configure all the devices to use the Network Time Protocol (NTP). The NTP configuration on each Cisco device is beyond the scope of this discussion. Refer to the product documentation at Cisco.com for specific information on NTP configuration.

Message

This is the text of the syslog message, along with some additional information about the process that generated the message. The syslog messages generated by Cisco IOS devices begin with a percent sign (%) and use the following format:

```
%FACILITY-SEVERITY-MNEMONIC: Message-text
```

Following is a description of each field:

- **FACILITY**—Refers to the source of the message, such as a hardware device, a protocol, or a module of the system software. Note that this FACILITY is Cisco specific and is only relevant within the message string. It is different from the facility defined in RFC 3164 for the syslog protocol.
- **SEVERITY**—This is similar to the severity defined in Table 4-2.
- **MNEMONIC**—This is a device-specific code that uniquely identifies the message.
- **Message-text**—This is a text string that describes the message and can contain details such as port numbers and network addresses.

Following is a sample syslog message generated by a Cisco IOS device:

```
*Mar  6 22:48:34.452 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
```

Note that the message begins with a special character (*) and that the timestamp includes the time-zone information. The message was generated by the LINEPROTO facility at severity 5 (Notice). The MNEMONIC UPDOWN along with the message-text describe the event.

The format of the syslog message generated by CatOS is slightly different from that generated by the IOS devices. Following is the format of the message generated by CatOS switches:

```
mm/dd/yyy:hh/mm/ss:facility-severity-MNEMONIC:Message-text
```

The syslog messages generated by a Cisco PIX Firewall begin with a percent sign (%) and are slightly different than the IOS syslog messages. Following is the format of syslog messages generated by a Cisco PIX Firewall:

```
%PIX-Level-Message_number: Message_text
```

For a complete list of the *Message_number* and *Message_text* and associated details , refer to the Cisco PIX Firewall System Log Messages section on the Cisco product documentation website (http://www.cisco.com/univercd/home/home.htm).

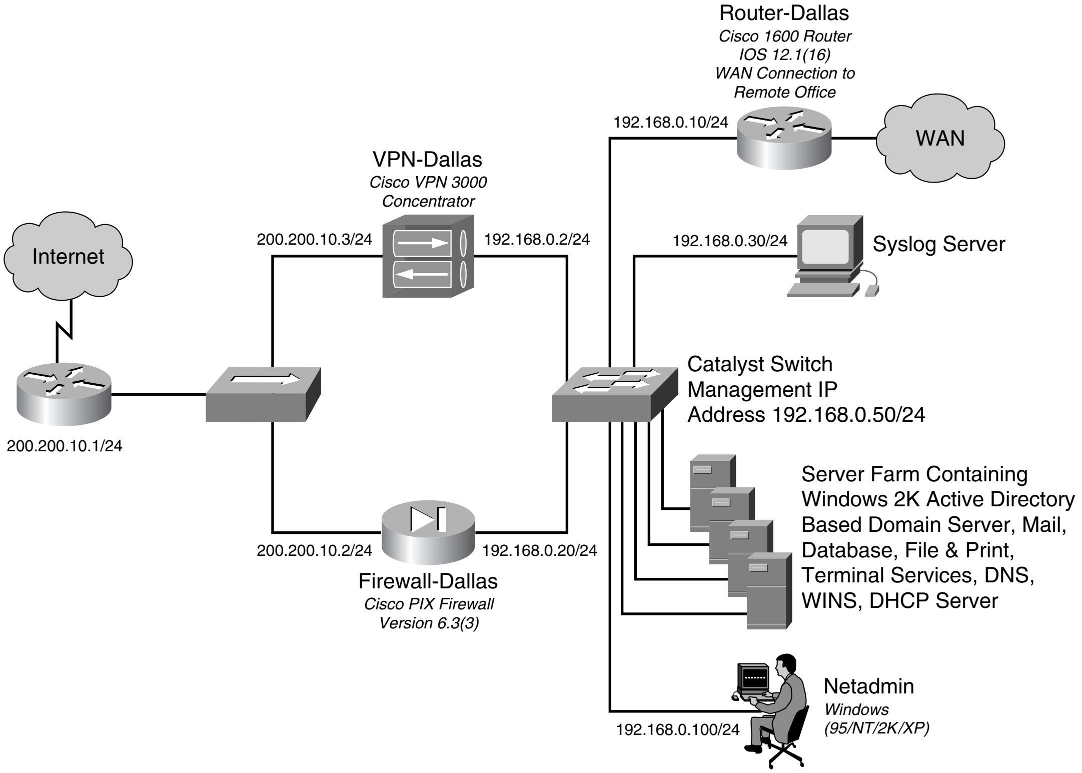
The syslog messages generated by Cisco VPN 3000 Concentrators follow the format of the IOS syslog messages, as discussed earlier in this section.

Deploying Syslog Servers

Consider a typical campus network of an organization consisting of routers, firewalls, VPN concentrators, and switches to interconnect the application servers to the users' workstations. Figure 4-1 shows a scaled-down version of the campus network of one such organization, ABC Investments.

To enable a centralized location to collect all the messages and alerts generated by various Cisco devices, the Netadmin has installed a syslog server. This server should be configured to accept, filter, and store syslog messages generated by the Cisco devices. Figure 4-2 illustrates LAN devices sending syslog messages to a central syslog server (with IP address 192.168.0.30). The following sections cover the step-by-step process of deploying the syslog server based on the choice of operating system.

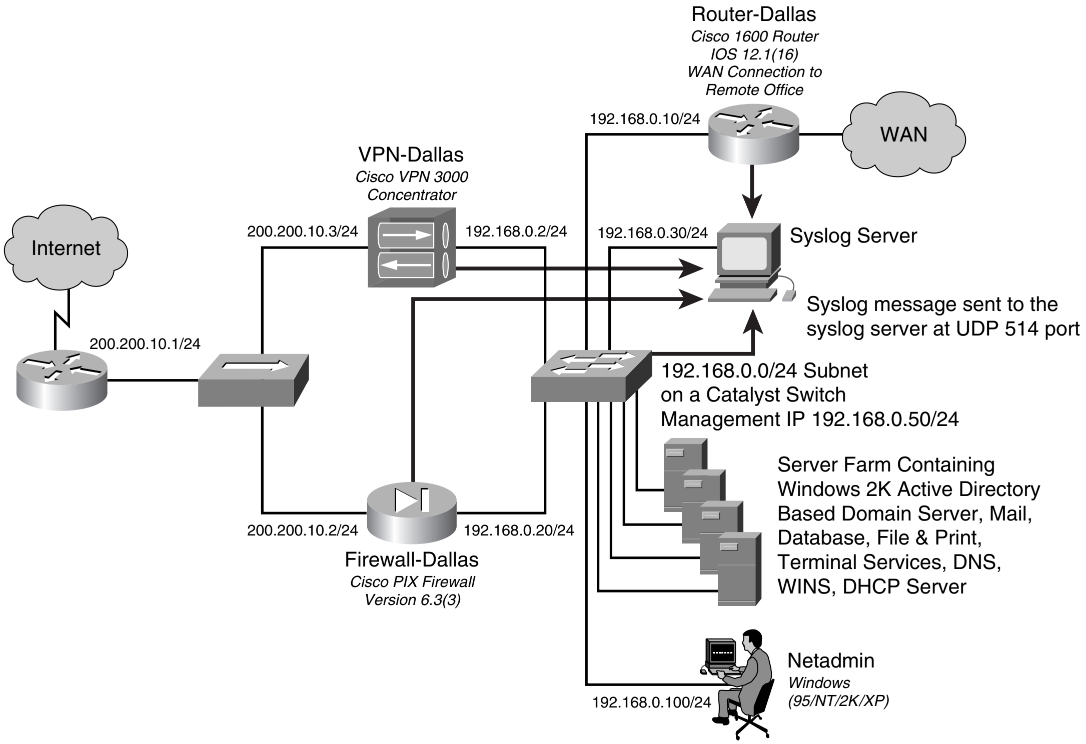
Figure 4-1 *Campus Network of ABC Investments*



Deploying the Default Syslog Daemon

To collect syslog messages generated by Cisco devices, many Netadmins might prefer to use the default syslog daemon that is included with the Linux operating system. Although it is well integrated with the operating system, the internal syslog server is not enabled for use as a network-based syslog server. To use the syslog daemon as a network-based syslog server, you must configure it through the /etc/syslog.conf file. Additionally, you must enable the syslog daemon to receive syslog messages from the network.

Figure 4-2 *Network Devices Sending Syslog Messages*



Configuring the Syslog Daemon

The /etc/syslog.conf file controls the configuration of the syslog daemon. This file contains the rules for sorting syslog messages. The rules specify the criteria by which to sort syslog messages on the basis of facility and severity levels, and to send them to destination log files. The default contents of the /etc/syslog.conf file are used to log the OS messages and should not be altered. To log messages from Cisco devices, simply append the new rules to the /etc/syslog.conf file. The syntax for specifying a rule is as follows:

```
facility.severity<Tab>destination-file-path
```

To log messages from Cisco devices, the possible values for facility are **local0** through **local7** and those for severity are **debug**, **info**, **notice**, **warning**, **err**, **crit**, **alert**, **emerg**, and **none**. The keyword **none** indicates no severity for the given facility. Although abbreviated, the keywords correspond to the severity levels listed in Table 4-2.

In addition, recall from Table 4-2 that debug is the highest level of severity. Specifying **debug** in the /etc/syslog.conf file results in all the messages (from severity debug to emergency) being logged. The use of **crit** results in logging all messages with a severity of critical, alert, and emergency, thereby excluding the higher levels (error, warning, notice, informational, and debug). To override this default behavior, you can use the special characters described in Table 4-3.

Table 4-3 *Special Characters in /etc/syslog.conf File**

Option	Description
,	Specifies multiple facilities with the same severity in one statement. The syntax is *facility1***,***facility2***.***severity*. Example: **local1,local5.debug**
;	Separates multiple pairs of *facility.severity* in the same line. Useful in conjunction with the **!** option.
*	Specifies all facilities or all severities.
none	Specifies no severity for the given facility.
=	Specifies only using the indicated severity level. The syntax is *facility.=severity*. For example, **local7.=debug** only logs level7 messages at the debug level and ignores the rest, such as info, notice, warning, and so on. Useful in overriding the default behavior of the syslog daemon to include lower severity messages.
!	Ignores the specified severity level, including the lower levels. The syntax is *facility.!severity*. Useful in conjunction with the **;** option. For example, **local7.*;local7.!err** logs all local7 messages but ignores messages with severity levels of error, critical, alert, and emergency.
destination-file-path	Specifies the location of the log file for storing the sorted messages. Instead of using the local file, you can also specify remote hosts using the @ option.
@	Specifies the host name or IP address of the remote syslog server. The syntax is *facility***.***severity***<Tab>**@*hostname*. If you use the host name, make sure that the host name is added in the /etc/hosts file.

*The contents of this table are derived from the syslog, sysklogd, and syslog.conf manual pages. Use the Linux **man** *command-name* for accessing the manual pages of any command. For example **man syslog** command will display the information about the syslog command.

You can create customized rules for sorting and storing various syslog messages based on the options listed in Table 4-3. For example, the following entry sends all the local7 messages to the file /var/log/router.log:

```
local7.debug  /var/log/router.log
```

By using the debug severity level, all the messages (from severity debug to emergency) are included. You can add multiple rules to the /etc/syslog.conf file. Depending on the configuration, a message can match multiple rules and hence can be sent to multiple log files.

NOTE Always use the Tab character instead of a space between the severity and destination fields of the /etc/syslog.conf file. Many Linux systems do not work with spaces in the /etc/syslog.conf file.

Example 4-1 contains several sample rules to be added to the /etc/syslog.conf file.

Example 4-1 *Sample Entries for the /etc/syslog.conf File*

```
# all message from all facilities will be sent to /var/log/allmessages.log file
*.debug                                      /var/log/allmessages.log
#
# Send all local4 messages to the /var/log/pix.log file
# By default Cisco PIX firewall uses local4 facility
local4.debug                                 /var/log/pix.log
#
# Send messages with facility local5 and severity level notice through emergency,
# to the /var/log/notice.log file
local5.notice                                /var/log/notice.log
#
# Only send messages with facility local4 and severity level of info
local4.=info                                 /var/log/onlyinfo.log
#
# Send all messages with facility local4 to the /var/log/errorandbelow.log file, but
# exclude messages with severity error, critical, alert and emergency
local4.*;local4.!err                         /var/log/errorandbelow.log
#
# Send all messages with facility local6 to the /var/log/allexcepterror.log file,
  and
# only exclude messages with severity level err
local6.*;local6.!=err                        /var/log/allexcepterror.log
```

By default, the syslog daemon only accepts local syslog messages. To enable the daemon to accept remote syslog messages, you must run the **syslogd** process in conjunction with the **-r** option. In Debian systems, the syslogd process is run through the /etc/init.d/sysklogd file. Using a text editor such as **vi**, you can edit the contents of the /etc/init.d/sysklogd file, as shown in Example 4-2.

Example 4-2 *Partial Contents of the /etc/init.d/sysklogd File*

```
#! /bin/sh
# /etc/init.d/sysklogd: start the system log daemon.
PATH=/bin:/usr/bin:/sbin:/usr/sbin
pidfile=/var/run/syslogd.pid
binpath=/sbin/syslogd
test -x $binpath || exit 0
# Options for start/restart the daemons
#   For remote UDP logging use SYSLOGD="-r"
#
#SYSLOGD=""
SYSLOGD="-r"
# ---OUTPUT SUPPRESSED---
```

Running the Syslog Daemon

After editing the configuration files, you must restart the syslog daemon. On Debian Linux machines, you can use the **init** script, as follows:

```
root@linuxbox:~# /etc/init.d/sysklogd restart
Restarting system log daemon: syslogd.
```

The syslog daemon is now ready for use as a network syslog server. To verify the operation of the syslog daemon, use the **ps** command, in conjunction with **grep**, as follows:

```
root@linuxbox:~# ps -ef | grep syslog
root      5750     1  0 19:45 ?        00:00:00 /sbin/syslogd -r
```

To verify that the syslog daemon is listening for remote syslog messages on the default UDP port of 514, use the **netstat** command, as follows:

```
root@linuxbox:~# netstat -na | grep 514
udp        0      0 0.0.0.0:514             0.0.0.0:*
```

The line entry beginning with udp indicates that the system is listening at UDP port 514.

If you make changes in the /etc/syslog.conf file, you can instruct the syslog daemon to reload the file, without restarting the entire syslog daemon, using the **kill** command, as follows:

```
root@linuxbox:~# kill -HUP `cat /var/run/syslogd.pid`
```

You can also use the **kill -1 `cat /var/run/syslog.pid`** command to get the same results.

To view the syslog messages, you can use the Linux system commands such as **cat**, **tail**, and **head**. For example, to view the last five syslog messages in the /var/log/pix.log file, use the **tail -n 5 /var/log/pix.log** command, as shown in Example 4-3.

Example 4-3 *Viewing Syslog Messages Using the* **tail** *Command*

```
root@linuxbox:~# tail -n 5 /var/log/pix.log
Apr 16 16:03:01 192.168.0.20 Apr 16 2005 15:37:27: %PIX-7-710002: TCP access
  permitted
from 192.168.0.150/20184 to inside:192.168.0.20/telnet
Apr 16 16:03:03 192.168.0.20 Apr 16 2005 15:37:29: %PIX-6-605005: Login permitted
  from
192.168.0.150/20184 to inside:192.168.0.20/telnet for user ""
Apr 16 16:03:07 192.168.0.20 Apr 16 2005 15:37:33: %PIX-7-111009: User 'enable_15'
executed cmd: show logging
Apr 16 16:04:37 192.168.0.20 Apr 16 2005 15:39:03: %PIX-7-111009: User 'enable_15'
   executed cmd: show running-config | inc logg
Apr 16 16:04:43 192.168.0.20 Apr 16 2005 15:39:09: %PIX-6-302010: 0 in use, 0 most
  used
root@linuxbox:~#
```

To view the messages in real time, use the **tail** command with **-f** option. In this case, the command would be **tail -f /var/log/pix.log**.

Deploying a Linux-Based Syslog-ng Server

As discussed earlier, Linux has a preinstalled syslog server, called Syslogd, that is an integral part of the operating system. However, the Syslogd daemon is older and suffers from the following limitations:

- **Facility**—The facilities labels (local0 to local7) are too general and are used by many programs. Such generic labels do not reflect the real facility that is generating the messages. For example, while the facility code 0 clearly indicates kernel messages, the facility code 20 (local4) does not indicate a particular facility and can be potentially used by any Cisco device.
- **Filtering**—Because all external programs are crowded together in the eight available local use facilities, many of the messages would end up using the same facility code. In such cases, selecting or filtering the messages from different devices becomes difficult. This makes it difficult to find the necessary information in the large number of log messages.

Several open source and commercial projects have tried to develop alternatives to the original syslog daemon. Some of them are Syslog-ng, SDSC Syslog, and Secure Syslog. Of these three, Syslog-ng, by Balazs Scheidler, is the next generation of syslog and offers the following advantages:

- **Filtering**—Syslog-ng can filter messages based on the contents of messages in addition to the priority/facility pair. This enables the Netadmin to log messages that are generated by each Cisco device to its own log file.
- **Ports**—Syslog-ng can use both Transmission Control Protocol (TCP) and UDP. This feature is useful for logging messages from devices, such as Cisco PIX Firewalls, that provide options to use syslog over TCP. Using syslog over TCP provides reliability because TCP is a connection-oriented protocol.
- **Long host name format**—The relay function offered by Syslog-ng allows syslog messages to traverse multiple Syslog-ng servers. In such cases, the long host name format, which records every intermediate Syslog-ng server, makes it easy to find the originating host and chain of forwarding hosts, even if a log message traverses several computers.
- **Active development and support**—Syslog-ng's development is ongoing, and it enjoys communitywide popularity and support.

Because Syslog-ng offers more options and flexibility to the Netadmin, the following sections discuss the details of deploying a Syslog-ng server.

Installing the Syslog-ng Daemon

The steps involved in installing a Syslog-ng daemon on a Linux server are as follows:

Step 1 Log in to the Linux machine using suitable login credentials.

Step 2 Open a web browser and download the Syslog-ng source file from http://www.balabit.com. The source file is in a compressed tar file (for example, syslog-ng-1.6.5.tar.gz). Additionally, download the source files for libol, which is the support library for Syslog-ng. (An example is libol-0.3.14.tar.gz.)

Step 3 Unpack the libol distribution by using the **tar xvfz libol-*x.x.xx*.tar.gz** command, where ***x.x.xx*** indicates the version number. This creates a directory named libol-x.xx, where the source for libol is unpacked. For the example shown in Step 2, the directory name is libol-0.3.14.

Step 4 Enter the libol-*x.x.xx* directory using the **cd libol-*x.x.xx*** command.

Step 5 Enter the following three commands to compile the source code:

```
./configure
make
make install
```

Step 6 After installing the libol package, change the working directory back to the one that contains the Syslog-ng source files. Unpack the distribution by using the **tar xvfz syslog-ng-*x.xx*.tar.gz** command, where ***x.xx*** stands for the version number. This creates a directory named syslog-ng-x.xx, where the source files for Syslog-ng are unpacked. For the example shown in Step 2, the directory name is syslog-ng-1.6.5.

Step 7 Enter the syslog-ng-*x.xx* directory using the **cd syslogng-*x.xx*** command.

Step 8 Enter the following three commands to compile the source code:

```
./configure
make
make install
```

The Syslog-ng daemon is now ready for configuration. Example 4-4 shows the commands that are used in the installation process.

Example 4-4 *Syslog-ng Installation*

```
[root@linuxbox]# tar zxvf libol-0.3.14.tar.gz
[root@linuxbox]# cd libol-0.3.14/
[root@linuxbox libol-0.3.14]# ./configure
[root@linuxbox libol-0.3.14]# make
[root@linuxbox libol-0.3.14]# make install
[root@linuxbox libol-0.3.14]# cd ..
[root@linuxbox]# tar xvfz syslog-ng-1.6.5.tar.gz
[root@linuxbox  root]# cd  syslog-ng-1.6.5
[root@linuxbox syslog-ng-1.6.5]# ./configure
[root@linuxbox syslog-ng-1.6.5]# make
[root@linuxbox syslog-ng-1.6.5]# make install
```

NOTE Debian users can avoid all the steps listed in this section and install Syslog-ng by using the **apt-get install syslog-ng** command.

Configuring the Syslog-ng Daemon

The Syslog-ng daemon is configured through the /etc/syslog-ng file. The following five components are used to configure the syslog-ng.conf file:

- options
- source
- destination
- filter
- log

Options

Syslog-ng.conf uses the **options** parameter to define global options for the Syslog-ng daemon. The command syntax is as follows:

```
options { option1(value); option2(value); ... };
```

Table 4-4 provides a partial list of options.

Table 4-4 *Partial List of Global Options in Syslog-ng*

Option Name	Accepted Values	Description
sync()	Number	The number of lines buffered before being written to the file.
create_dirs()	**yes** or **no**	Enables or disables directory creation; helpful when using macros in the file destination drivers.
chain_hostnames()	**yes** or **no**	Enables or disables the chained host name format.
long_hostnames()	**yes** or **no**	Alias for **chain_hostnames**.
keep_hostname()	**yes** or **no**	Replaces the host name in the message with its DNS name. If **keep_hostname** is **yes** and **chain_hostnames** is **yes**, the sender's name is appended to the DNS host name; otherwise the name is replaced.
use_dns()	**yes** or **no**	Enables or disables DNS usage. Syslog-ng blocks on DNS queries, so enabling DNS can lead to a denial of service (DoS) attack. To prevent DoS attacks, protect your Syslog-ng network endpoint with firewall rules, and make sure that all hosts that can get to Syslog-ng are resolvable.
use_fqdn()	**yes** or **no**	Adds a fully qualified domain name (FQDN) instead of a short host name.

Example 4-5 shows a sample snippet for the options components of the /etc/syslog-ng.conf file. This code prepares the Syslog-ng daemon to be used as a central syslog server for Cisco devices.

Example 4-5 *Syslog-ng.conf—Options Components*

```
options {
        chain_hostnames(yes);
        keep_hostname(yes);
        use_fqdn(yes);
use_dns(no)
        sync(0);
};
```

Source

The **source** statement defines one or more source categories used by the Syslog-ng daemon to collect messages. The /etc/syslog-ng.conf file refers to these sources as source-drivers. The command syntax for declaring all the sources is as follows:

```
source identifier { source-driver(params); source-driver(params); ... };
```

The *identifier* is a text string that uniquely identifies the source. Table 4-5 provides a partial list of source-drivers.

Table 4-5 *Partial List of Source-Drivers in Syslog-ng*

Source-Driver Name	Description
internal	Indicates messages that are generated internally in Syslog-ng
unix-stream	Opens the specified UNIX socket in SOCK_STREAM mode and listens for messages
unix-dgram	Opens the specified UNIX socket in SOCK_DGRAM mode and listens for messages
udp	Listens on the specified UDP port for messages
tcp	Listens on the specified TCP port for messages

Note the last two entries in Table 4-5. The UDP and TCP source-drivers enable the Syslog-ng daemon to act as a central syslog server. These source-drivers instruct the daemon to accept messages through the network.

Example 4-6 shows a sample snippet for the source components of the syslog-ng.conf file. The code (with identifier **s_cisconetwork**) prepares the Syslog-ng daemon to get syslogs sent by Cisco devices through the network at the default UDP port of 514.

Example 4-6 *Syslog-ng.conf—Source Components*

```
# source s_cisconetwork will listen on default UDP514
source s_cisconetwork {
     udp();
};
```

destination

The **destination** statement is used by the daemon to direct the syslog messages after filtering. Similar to sources, destinations use one or more destination-drivers to define message handling.

The command syntax for declaring the all the sources is as follows:

```
destination identifier { destination-driver(params); destination-driver(params);
... };
```

The identifier is a text string that uniquely identifies the destination list. Table 4-6 provides a partial list of destination-drivers.

Table 4-6 *Partial List of Destination-Drivers in Syslog-ng*

Destination-Driver Name	Description
file	Writes messages to the given file; this is the most commonly used option.
udp	Sends messages to the specified host and UDP port; this enables the syslog server to act as a relay server.
tcp	Sends messages to the specified host and TCP port; this enables the syslog server to act as a relay server.
program	Launches the specified program in the background and sends messages to its standard input; useful for incorporating **syslog-ng** with external scripts.

The first entry in Table 4-6, the file driver, is one of the most important destination-drivers in Syslog-ng. It allows you to include macros to automatically create new files based on the syslog message content. Note that this functionality requires the use of the **create_dirs(yes)** option in the destination-driver statement. The macros are included by prefixing the macro name with a dollar sign ($) (such as $HOSTS and $LEVEL).

For example, the following statement uses the $HOST macro in the file destination-driver:

```
destination hosts { file("/var/log/host/$HOST" create_dirs(yes)); };
```

This creates a new log file for each of the hosts that sends a network message to this Syslog-ng daemon. The syslog messages sent by the host Router-Dallas are stored in the log file /var/log/host/router-dallas. If the router-dallas file does not exist, it is automatically created. Table 4-7 provides a complete list of macros that are available for the file destination-driver. As shown in this example, these macros provide highly flexible methods of handling syslog messages. A Netadmin can control the logging of syslog based on the host name, facility, severity, date, and timestamp of the syslog messages generated by Cisco devices.

Table 4-7 *Available Macros in the File Destination-Driver*

Name	Description
FACILITY	The name of the facility that the message is tagged as coming from.
PRIORITY or LEVEL	The priority or the severity level of the message.
TAG	The priority and facility encoded as a 2-digit hexadecimal number.
DATE	Date of the transaction.
FULLDATE	Long form of the date of the transaction.
ISODATE	Date in ISO format.
YEAR	The year the message was sent. Time expansion macros can either use the time specified in the log message (for example, the time the log message is sent) or the time the message was received by the log server. This is controlled by the **use_time_recvd()** option.
MONTH	The month the message was sent.
DAY	The day of the month the message was sent.
WEEKDAY	The three-letter name of the day of the week the message was sent (for example, **Thu**).
HOUR	The hour of the day the message was sent.
MIN	The minute the message was sent.
SEC	The second the message was sent.
FULLHOST	The full host name of the system that sent the log.
HOST	The name of the source host where the message originated. If the message traverses several hosts, and **chain_hostnames()** is set to **yes**, the name of the first host is used.
PROGRAM	The name of the program that the message was sent by.
MSG or MESSAGE	Message contents.

TIP Instead of grouping by host name, you can also group files by time, such as day, date, or weekday. For example by using the DATE macro, messages are sorted by their date of creation. Consequently, by the end of the year, you will have 365 different files. Additionally, each file will contain messages generated by all the devices on the given day. This chronological grouping of all the messages helps Netadmins to correlate events across multiple devices. The command syntax is as follows:

```
destination hosts { file("/var/log/host/$DATE" create_dirs(yes)); };
```

Along with using the macros listed in Table 4-7, the file destination-driver also allows the use of local options that override the global options listed in the beginning of the syslog-ng.conf file. Table 4-8 shows a partial list of these options.

Table 4-8 *Partial List of Options for File Destination-Driver*

Name	Type	Description
owner()	String	Sets the owner of the created filename to the one specified. The default is **root**.
group()	String	Sets the group of the created filename to the one specified. The default is **root**.
perm()	Number	Indicates the permission mask of the file if it is created by Syslog-ng. The default is **0600**.
dir_perm()	Number	Indicates the permission mask of directories created by Syslog-ng. Log directories are only created if a file, after macro expansion, refers to a nonexisting directory, and dir creation is enabled using **create_dirs()**. The default is **0600**.
create_dirs()	**yes** or **no**	Enables the creation of nonexisting directories. The default is **no**.

Example 4-7 shows a sample snippet for the destination components of the /etc/syslog-ng.conf file. This code instructs the Syslog-ng daemon to create separate log files for each host that sends syslog messages. The second part of the code (with the **d_cisco_facility** identifier) instructs the Syslog-ng daemon to create separate files for each message based on the facility code. The third part of the code (with the **d_cisco_severity** identifier) instructs the Syslog-ng daemon to create separate files for each message based on the severity code.

Example 4-7 *Syslog-ng.conf—Destination Components*

```
destination d_hosts {
    file("/var/log/HOSTS/$HOST.log"
    create_dirs(yes));
};
destination d_cisco_facility {
    file("/var/log/FACILITY/$FACILITY.log"
    create_dirs(yes));
};
destination d_cisco_severity {
    file("/var/log/LEVEL/$LEVEL.log"
    create_dirs(yes));
};
```

filter

The **filter** statement is used by Syslog-ng to route the syslog messages. You can use a Boolean expression to allow a message to pass through the filter. The syntax is as follows:

```
filter identifier { expression; };
```

The identifier is a text string that uniquely identifies the filters in the log statements. An expression can contain parentheses; the Boolean operators AND, OR, and NOT; and any of the functions listed in Table 4-9.

Table 4-9 *Available Filter Functions in Syslog-ng*

Filter Function Name	Description
facility(*facility*[*,facility*]**)**	Matches messages having one of the listed facility codes.
level(*pri*[*,pri1..pri2*[*,pri3*]]**)**	Matches messages based on priority or severity level. Multiple priorities are separated by commas, and range of priorities is specified by listing the upper and lower priorities separated by two periods.
program(*regexp***)**	Matches messages by using a regular expression against the program name field of log messages.
host(*regexp***)**	Matches messages by using a regular expression against the host name field of log messages.
match()	Tries to match a regular expression to the message itself. This feature is useful for customized filtering based on a specific text string in a message.
filter()	Calls another filter rule and evaluates its value.

Example 4-8 shows a sample snippet for the filter components of the /etc/syslog-ng.conf file. The first four filter statements sort the messages based on their facility code. For this filter to work as intended, the Netadmin must configure all the routers to use facility at local2, switches at local3, firewall at local4, and VPN concentrators at local5. If the Netadmin needs to catch all messages with severity level error and above, he can use the last filter (with identifier **f_errandabove**).

Example 4-8 *Syslog-ng.conf—Filter Components*

```
filter      f_router       { facility(local2); };
filter      f_switch       { facility(local3); };
filter      f_firewall     { facility(local4); };
filter      f_vpnbox       { facility(local5); };
filter      f_errandabove  { level(err..emerg);};
```

log

The **log** statement is used to combine the source, filter, and destination components. The syntax is as follows:

```
log { source(s1); source(s2); ...
      filter(f1); filter(f2); ...
      destination(d1); destination(d2); ...
      flags(flag1[, flag2...]); };
```

Messages coming from any of the listed sources that match the listed filters are sent to all listed destinations. Because log statements are processed in the order they appear in the config file, a single log message might be sent to the same destination several times. This default behavior can be changed by using the **flag** parameters listed in Table 4-10.

Table 4-10 *Log Statement Flags*

Flag	Description
final	This flag means that the processing of log statements ends here. Note that this doesn't necessarily mean that matching messages will be stored once because they can be matching log statements processed prior to the current one.
fallback	This flag makes a log statement "fall back." A fallback statement means that only messages not matching any nonfallback log statements are dispatched.
catchall	This flag means that the source of the message is ignored; only the filters are taken into account when matching messages.

Example 4-9 shows a sample snippet for the log components of the /etc/syslog-ng.conf file.

Example 4-9 *Syslog-ng.conf—Log Components*

```
log { source(s_cisconetwork); filter(f_errandabove); destination(d_cisco_severity);
  };
log { source(s_cisconetwork); filter(f_router); destination(d_cisco_facility); };
log { source(s_cisconetwork); destination(d_hosts); };
```

The first log statement does the following:

1 Listens for all network messages at UDP port 514.

2 Filters to only select messages with a severity level between error and emergency.

3 Sends the filtered messages to the respective log files. The name of destination file matches the severity level of the message. For example, the messages with severity level "*error*" will be sent to the "/var/log/LEVEL/*err.log*" file.

The second log statement does the following:

1 Listens for all network messages at UDP port 514.

2 Filters to only select messages with a facility of local2.

3 Sends these filtered messages to respective log files. The name of the destination file matches the facility level of the message. For example messages from facility level "*Local 2*" will be sent to the "*/var/log/FACILITY/local2.log*" file.

The third log statement does the following:

1 Listens for all network messages at UDP port 514.

2 Sends these messages to the respective log files. The name of destination file matches the name of the host who generated the message. For example, messages from the host "*router-dallas*" will be sent to "*/var/log/HOSTS/router-dallas.log*" file.

CAUTION Deleting the default statements in the original /etc/syslog-ng.conf is not recommended. To prepare the Syslog-ng daemon as a central syslog server, just add the relevant code snippets in the respective sections of the syslog-ng.conf file. Also, it is good administrative practice to save the original /etc/syslog-ng/syslog-ng.conf file using the following command:

```
linuxbox:~# mv /etc/syslog-ng/syslog-ng.conf  /etc/syslog-ng/syslog-
  ng.conf.orig
```

Example 4-10 shows the working copy of the /etc/syslog-ng.conf file after editing. The new configuration is highlighted throughout the file. Note that this is the default /etc/syslog-ng.conf file that is installed with the **apt-get install syslog-ng** command on a Debian 3.0 stable release.

Example 4-10 *Edited Copy of the /etc/syslog-ng/syslog-ng.conf File*

```
# Syslog-ng configuration file, compatible with default Debian syslogd
# installation. Originally written by anonymous (I can't find his name)
# Revised, and rewrited by me (SZALAY Attila <sasa@debian.org>)
# First, set some global options.
#options { long_hostnames(off); sync(0); };
# NOTE THE NEW OPTIONS LIST
options {
chain_hostnames(yes); keep_hostname(yes);
use_fqdn(yes);
use_dns(no);
sync(0);
};
#
# This is the default behavior of sysklogd package
# Logs may come from unix stream, but not from another machine.
#
source src { unix-dgram("/dev/log"); internal(); };
#
# If you wish to get logs from remote machine you should uncomment
# this and comment the above source line.
#
# source src { unix-dgram("/dev/log"); internal(); udp(); };
# the following source driver will enable listening for network mssg on udp514
source s_cisconetwork {udp(); };
# After that set destinations.
# First some standard logfile
#
destination authlog { file("/var/log/auth.log" owner("root") group("adm")
 perm(0640)); };
destination syslog { file("/var/log/syslog" owner("root") group("adm") perm(0640)); };
```

Example 4-10 *Edited Copy of the /etc/syslog-ng/syslog-ng.conf File (Continued)*

```
destination cron { file("/var/log/cron.log" owner("root") group("adm") perm(0640)); };
destination daemon { file("/var/log/daemon.log" owner("root") group("adm")
  perm(0640)); };
destination kern { file("/var/log/kern.log" owner("root") group("adm") perm(0640)); };
destination lpr { file("/var/log/lpr.log" owner("root") group("adm") perm(0640)); };
destination mail { file("/var/log/mail.log" owner("root") group("adm") perm(0640)); };
destination user { file("/var/log/user.log" owner("root") group("adm") perm(0640)); };
destination uucp { file("/var/log/uucp.log" owner("root") group("adm") perm(0640)); };

# This files are the log come from the mail subsystem.
#
destination mailinfo { file("/var/log/mail.info" owner("root") group("adm")
  perm(0640)); };
destination mailwarn { file("/var/log/mail.warn" owner("root") group("adm")
  perm(0640)); };
destination mailerr { file("/var/log/mail.err" owner("root") group("adm")
  perm(0640)); };
# Logging for INN news system
#
destination newscrit { file("/var/log/news/news.crit" owner("root") group("adm")
  perm(0640)); };
destination newserr { file("/var/log/news/news.err" owner("root") group("adm")
  perm(0640)); };
destination newsnotice { file("/var/log/news/news.notice" owner("root")
  group("adm") perm(0640)); };
# Some `catch-all' logfiles.
#
destination debug { file("/var/log/debug" owner("root") group("adm") perm(0640));
  };
destination messages { file("/var/log/messages" owner("root") group("adm")
  perm(0640)); };
# The root's console.
#
destination console { usertty("root"); };
# Virtual console.
#
destination console_all { file("/dev/tty8"); };
# The named pipe /dev/xconsole is for the nsole' utility.  To use it,
# you must invoke nsole' with the -file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
destination xconsole { pipe("/dev/xconsole"); };
destination ppp { file("/var/log/ppp.log" owner("root") group("adm") perm(0640)); };
# following destination drivers were added for cisco devices
destination d_hosts {
    file("/var/log/HOSTS/$HOST.log"
        create_dirs(yes));
        };
destination d_cisco_facility {
    file("/var/log/FACILITY/$FACILITY.log"
    create_dirs(yes));
};
```

continues

Example 4-10 *Edited Copy of the /etc/syslog-ng/syslog-ng.conf File (Continued)*

```
destination d_cisco_severity {
    file("/var/log/LEVEL/$LEVEL.log"
    create_dirs(yes));
};

# Here's come the filter options. With this rules, we can set which
# message go where.
filter f_authpriv { facility(auth, authpriv); };
filter f_syslog { not facility(auth, authpriv); };
filter f_cron { facility(cron); };
filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail); };
filter f_user { facility(user); };
filter f_uucp { facility(uucp); };
filter f_news { facility(news); };
filter f_debug { not facility(auth, authpriv, news, mail); };
filter f_messages { level(info .. warn)
    and not facility(auth, authpriv, cron, daemon, mail, news); };
filter f_emergency { level(emerg); };
filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_crit { level(crit); };
filter f_err { level(err); };
filter f_cnews { level(notice, err, crit) and facility(news); };
filter f_cother { level(debug, info, notice, warn) or facility(daemon, mail); };
filter ppp { facility(local2); };
# following filters were added for cisco devices
filter      f_router       { facility(local2); };
filter      f_switch        { facility(local3); };
filter      f_firewall        { facility(local4); };
filter      f_vpnbox       { facility(local5); };
filter      f_errandabove  { level(err..emerg); };

log { source(src); filter(f_authpriv); destination(authlog); };
log { source(src); filter(f_syslog); destination(syslog); };
#log { source(src); filter(f_cron); destination(cron); };
log { source(src); filter(f_daemon); destination(daemon); };
log { source(src); filter(f_kern); destination(kern); };
log { source(src); filter(f_lpr); destination(lpr); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(user); };
log { source(src); filter(f_uucp); destination(uucp); };
log { source(src); filter(f_mail); filter(f_info); destination(mailinfo); };
log { source(src); filter(f_mail); filter(f_warn); destination(mailwarn); };
log { source(src); filter(f_mail); filter(f_err); destination(mailerr); };
log { source(src); filter(f_news); filter(f_crit); destination(newscrit); };
log { source(src); filter(f_news); filter(f_err); destination(newserr); };
log { source(src); filter(f_news); filter(f_notice); destination(newsnotice); };
log { source(src); filter(f_debug); destination(debug); };
```

Example 4-10 *Edited Copy of the /etc/syslog-ng/syslog-ng.conf File (Continued)*

```
log { source(src); filter(f_messages); destination(messages); };
log { source(src); filter(f_emergency); destination(console); };
#log { source(src); filter(f_cnews); destination(console_all); };
#log { source(src); filter(f_cother); destination(console_all); };

log { source(src); filter(f_cnews); destination(xconsole); };
log { source(src); filter(f_cother); destination(xconsole); };
log { source(src); filter(ppp); destination(ppp); };
# following logs were added for cisco devices
log { source(s_cisconetwork); filter(f_errandabove); destination(d_cisco_severity);
 };
log { source(s_cisconetwork); filter(f_router); destination(d_cisco_facility); };
log { source(s_cisconetwork); destination(d_hosts); };
# config ends here
```

Starting the Syslog-ng Daemon

After configuring the syslog-ng.conf file to deploy a Linux-based central syslog server, you must restart the daemon. The Syslog-ng daemon can be started or stopped using the init script, as follows:

```
/etc/init.d/syslog-ng {start | stop | restart | reload | force-reload}
```

The following example shows the command that restarts the daemon:

```
linuxbox:~# /etc/init.d/syslog-ng restart
```

Viewing the Logs

All the logs are stored in the location defined by the file destination-driver. Use the **tail** command to view the latest messages added to a particular log file. Examples are as follows:

```
Linuxbox:~# tail /var/log/HOSTS/router-dallas.log
Linuxbox:~# tail /var/log/LEVEL/err.log
Linuxbox:~# tail /var/log/FACILITY/local2.log
```

Configuring a Windows-Based Syslog Server

The MS-Windows–based servers have a syslog-like feature called the Event Viewer. However, the message format used by the Event Viewer is proprietary to Microsoft and is not compatible with the UNIX syslog. To run a syslog server on MS-Windows machines, you need to install a third-party utility. One of the most popular syslog servers for Windows is Kiwi Syslogd Server. Some of the outstanding features of Kiwi Syslogd Server are as follows:

- It is available as freeware, allowing users to run it indefinitely.
- It runs as a service in the background.

- It offers a GUI for easy management.
- It uses both TCP and UDP ports, thus enabling the server to accept PIX TCP syslogs.
- Its built-in syslog viewer displays messages in real time.
- It features automatic log-file archiving based on a custom schedule.
- It shows syslog statistics with a graph of syslog trends (last 24 hours/last 60 minutes).
- It features a messages-per-hour alarm with sound or e-mail notification.
- It has an alarm with sound or e-mail notification if the log file size exceeds a threshold.
- It can send daily e-mails of syslog traffic statistics.

Deploying a Kiwi Syslog Server consists of following steps:

- Installing the syslog server
- Configuring the syslog server
- Starting the syslog server
- Viewing the syslog messages from the clients

Installing the Syslog Server

Download and save the service version of the Kiwi Syslog Daemon from http://www.kiwisyslog.com. The service version runs syslog as a service in the background instead of as an application on the desktop. At the time of this writing, the current stable version is 7.14, and the installation filename is Kiwi_Syslogd_Service.exe.

Begin the installation process by double-clicking the downloaded file and following the default values when prompted.

Configuring the Syslog Server

After the installation is complete, follow these steps to configure the Kiwi Syslog Server:

Step 1 Choose **Start** > **Programs** > **Kiwi Enterprises** > **Kiwi Syslog Daemon** > **Kiwi Syslog Daemon** to open the Kiwi Syslog Service Manager window (see Figure 4-3).

Step 2 Choose **Manage** > **Install the Syslogd service** to install the syslog service.

Step 3 After the syslog service is installed, the syslog server should be configured for archiving, e-mails, alarms, and TCP ports. To begin configuration, choose **File** > **Setup** to open the Kiwi Syslog Daemon Setup window, as shown in Figure 4-4.

Figure 4-3 *Kiwi Syslog Manager*

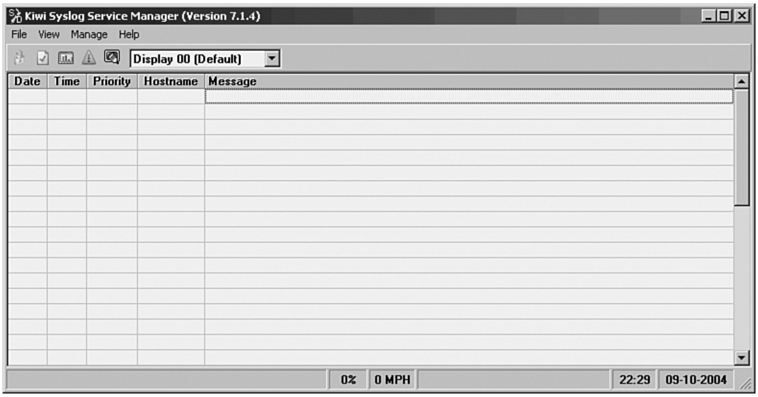
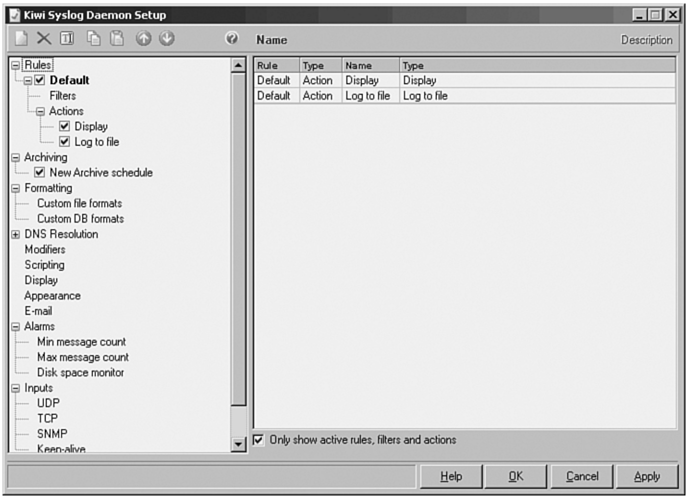


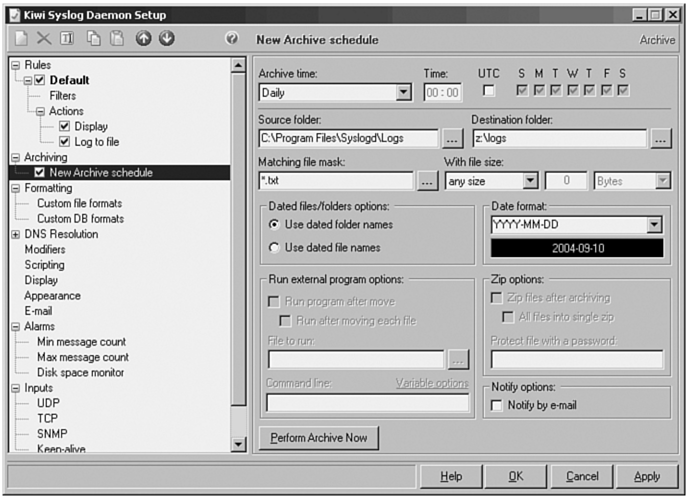
Figure 4-4 *Kiwi Syslog Setup*



Rule	Type	Name	Type
Default	Action	Display	Display
Default	Action	Log to file	Log to file

Step 4 In the left pane of the setup window, click **Archiving** and select the **New Archive schedule** check box. Netadmins can choose archive frequencies ranging from hourly to yearly (or customized). Your choice should be based on the network environment, frequency of system backups, and amount of logs generated. Figure 4-5 shows daily archiving. Also, note that the Destination folder field indicates the network drive z:\logs. Archiving to a network drive helps to avoid loss of data because of local device failures.

Figure 4-5 *Kiwi Syslog Setup—Archiving*



Archive time: Daily
Time: 00 : 00
UTC □
S ☑ M ☑ T ☑ W ☑ T ☑ F ☑ S ☑
Source folder: C:\Program Files\Syslogd\Logs
Destination folder: z:\logs
Matching file mask: *.txt
With file size: any size 0 Bytes

Dated files/folders options:
● Use dated folder names
○ Use dated file names

Date format: YYYY-MM-DD
2004-09-10

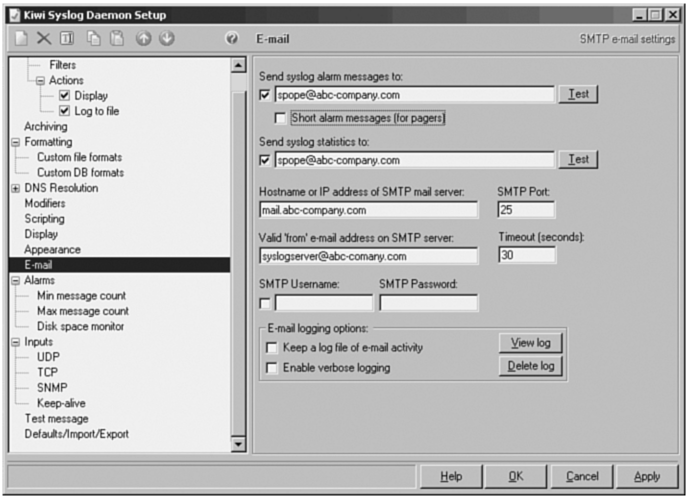
Run external program options:
□ Run program after move
□ Run after moving each file
File to run:
Command line: Variable options

Zip options:
□ Zip files after archiving
□ All files into single zip
Protect file with a password:

Notify options:
□ Notify by e-mail

Step 5 In the E-mail section of the setup window, configure the e-mail settings according to your organization's mail server settings. Figure 4-6 shows that the syslog alarm and statistics are sent to the e-mail address spope@abc-company.com, using the mail server mail.abc-company.com. The mail recipient would see the messages as coming from syslogserver@abc-company.com.

Figure 4-6 *Kiwi Syslog Setup—E-Mail Setup*



Send syslog alarm messages to:
☑ spope@abc-company.com
□ Short alarm messages (for pagers)

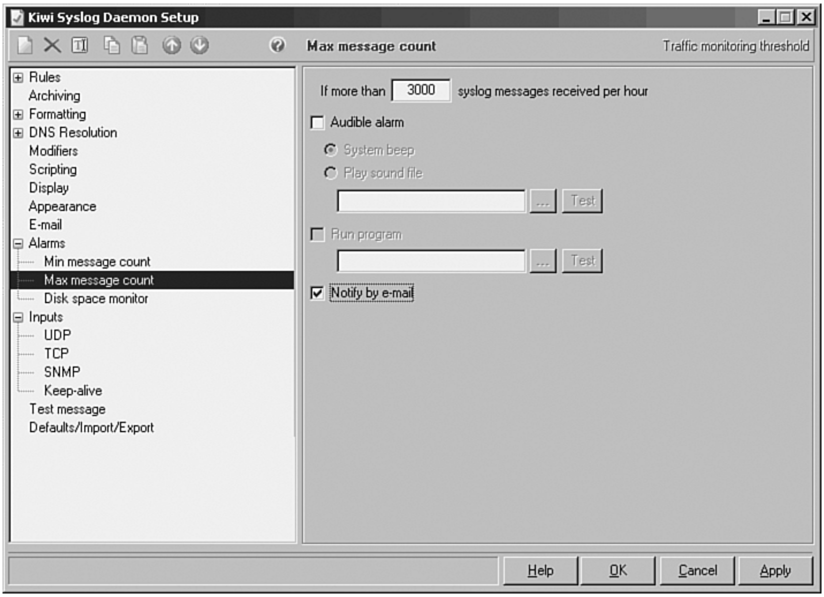
Send syslog statistics to:
☑ spope@abc-company.com

Hostname or IP address of SMTP mail server: mail.abc-company.com
SMTP Port: 25
Valid 'from' e-mail address on SMTP server: syslogserver@abc-comany.com
Timeout (seconds): 30
SMTP Username: □
SMTP Password:

E-mail logging options:
□ Keep a log file of e-mail activity
□ Enable verbose logging

Step 6 In the Alarms section of the setup window, enter the settings shown in Figure 4-7. You should set a limit for the number of messages received in an hour because an unusually high number can indicate a problem. However, the exact number depends on your network environment. Also, setting up the alarms for disk space (see Figure 4-8) notifies the Netadmin before the server stops working because of insufficient disk space. This option is useful for Cisco PIX Firewalls that use TCP syslogs. Cisco PIX Firewalls, using the TCP syslog feature, stop processing traffic if the hard drive on the syslog server is full. This is a security feature, but it can effectively cause a denial of service for legitimate users.

Figure 4-7 *Kiwi Syslog Setup—Message Count*



Kiwi Syslog Daemon Setup

Max message count

Traffic monitoring threshold

If more than 3000 syslog messages received per hour

☐ Audible alarm

● System beep

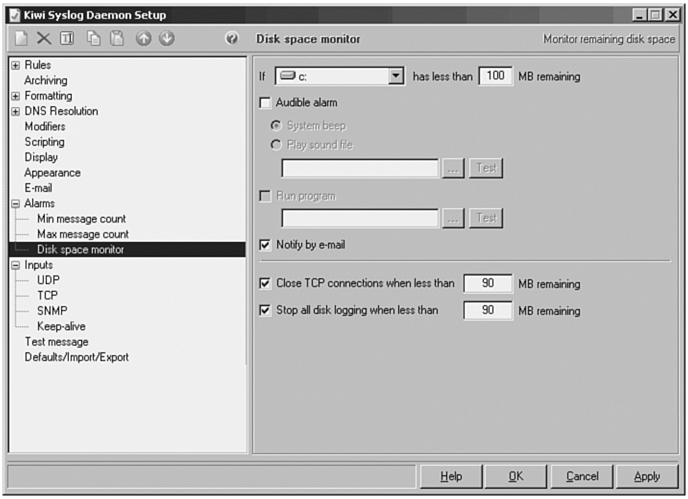
○ Play sound file

☐ Run program

☑ Notify by e-mail

Step 7 To configure the syslog server to listen over TCP ports, click **Inputs > TCP** in the setup window and enter the settings shown in Figure 4-9. Note that the TCP port 1468 is the default port used by PIX for TCP syslogs.

Figure 4-8 *Kiwi Syslog Setup—Disk Space*



If c: has less than 100 MB remaining

☐ Audible alarm

● System beep

○ Play sound file

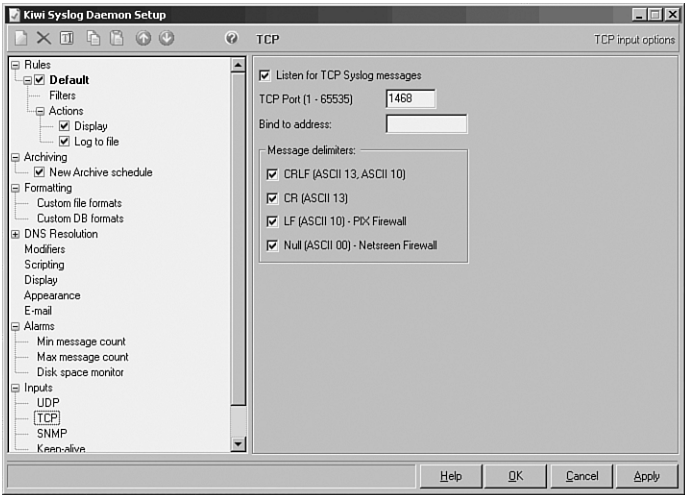
☐ Run program

☑ Notify by e-mail

☑ Close TCP connections when less than 90 MB remaining

☑ Stop all disk logging when less than 90 MB remaining

Figure 4-9 *Kiwi Syslog Setup—TCP*



☑ Listen for TCP Syslog messages

TCP Port (1 - 65535) 1468

Bind to address:

Message delimiters:

☑ CRLF (ASCII 13, ASCII 10)

☑ CR (ASCII 13)

☑ LF (ASCII 10) - PIX Firewall

☑ Null (ASCII 00) - Netsreen Firewall

Step 8 Now that the syslog server is configured, save the settings and close the setup window by clicking the **OK** button.

Starting the Syslog Server

Start the Syslog service by choosing **Manage > Start the Syslogd service**. The status bar briefly displays the message **The syslog server has been started**.

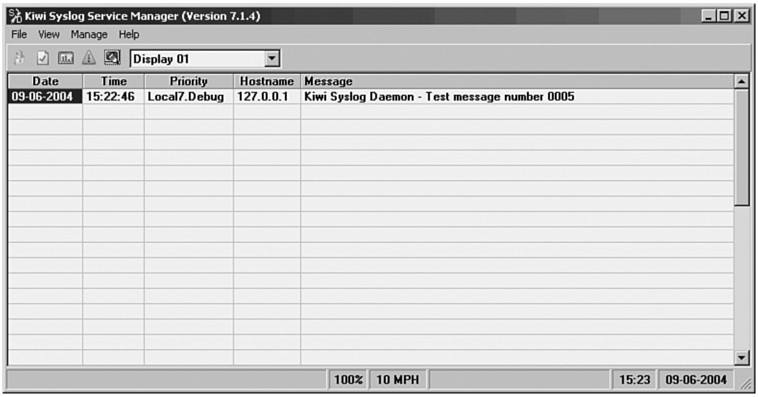
The server is now ready to accept the syslog messages.

Viewing Messages on the Syslog Server

To test the operation of the syslog server, choose **File > Send test message to localhost** in the Service Manager window.

The Service Manager window displays the message shown in Figure 4-10.

Figure 4-10 *Kiwi Syslog Manager*



Date	Time	Priority	Hostname	Message
09-06-2004	15:22:46	Local7.Debug	127.0.0.1	Kiwi Syslog Daemon - Test message number 0005

By default, the syslog server saves all logs to the following text file: C:\Program Files\Syslogd\Logs\SyslogCatchAll.txt.

To view the logs, open the SyslogCatchAll.txt file in a text editor. Example 4-11 shows the sample content of the SyslogCatchAll.txt file as viewed using Windows Notepad.

Example 4-11 *Contents of the SyslogCatchAll.txt File*

```
2005-04-16 21:50:56 Local7.Notice      192.168.0.10      9071: Apr 16 20:50:57.852
  PST: %SYS-5-CONFIG_I: Configured from console by vty0 (192.168.0.150)
2005-04-16 21:50:56 Local7.Notice      192.168.0.10      9072: Apr 16 20:50:58.388
  PST: %LINK-5-CHANGED: Interface Loopback0, changed state to administratively down
2005-04-16 21:50:58 Local7.Notice      192.168.0.10      9073: Apr 16 20:50:59.380
  PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
  down
2005-04-16 21:51:13 Local4.Debug      192.168.0.20      %PIX-7-710005: UDP request
  discarded from 192.168.0.151/33375 to inside:192.168.0.255/sunrpc
2005-04-16 21:51:17 Local4.Debug      192.168.0.20      %PIX-7-710005: UDP request
  discarded from 192.168.0.151/33375 to inside:192.168.0.255/sunrpc
2005-04-16 21:51:22 Local7.Notice      192.168.0.10      9074: Apr 16 20:51:23.588
  PST: %SYS-5-CONFIG_I: Configured from console by vty0 (192.168.0.150)
```

continues

Example 4-11 *Contents of the SyslogCatchAll.txt File (Continued)*

```
2005-04-16 21:51:23 Local7.Error      192.168.0.10      9075: Apr 16 20:51:24.771
  PST: %LINK-3-UPDOWN: Interface Loopback0, changed state to up
2005-04-16 21:51:23 Local7.Notice      192.168.0.10      9076: Apr 16 20:51:25.763
  PST: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
2005-04-16 21:58:12 Local7.Notice     192.168.0.10      9077: .Apr 16 20:58:12.142
  PST: %SYS-5-CONFIG_I: Configured from console by vty0 (192.168.0.150)
2005-04-16 21:58:45 Local7.Notice     192.168.0.10      9078: .Apr 17 04:58:45.596
  UTC: %SYS-5-CONFIG_I: Configured from console by vty0 (192.168.0.150)
2005-04-16 21:58:49 Local4.Debug      192.168.0.20      %PIX-7-710005: UDP request
  discarded from 192.168.0.151/33382 to inside:192.168.0.255/sunrpc
2005-04-16 21:58:57 Local4.Debug      192.168.0.20      %PIX-7-710005: UDP request
  discarded from 192.168.0.151/33382 to inside:192.168.0.255/sunrpc
2005-04-16 21:59:07 Local4.Debug      192.168.0.20      %PIX-7-710005: UDP request
  discarded from 192.168.0.151/33382 to inside:192.168.0.255/sunrpc
2005-04-16 21:59:09 Local7.Notice     192.168.0.10      9079: .Apr 17 04:59:08.748
  UTC: %LINK-5-CHANGED: Interface Loopback0, changed state to administratively down
2005-04-16 21:59:09 Local7.Notice     192.168.0.10      9080: .Apr 17 04:59:09.740
  UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
  down
2005-04-16 21:59:11 Local7.Notice     192.168.0.10      9081: .Apr 17 04:59:11.324
  UTC: %SYS-5-CONFIG_I: Configured from console by vty0 (192.168.0.150)
2005-04-16 21:59:13 Local7.Error      192.168.0.10      9082: .Apr 17 04:59:12.998
  UTC: %LINK-3-UPDOWN: Interface Loopback0, changed state to up
2005-04-16 21:59:13 Local7.Notice     192.168.0.10      9083: .Apr 17 04:59:13.990
  UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
```

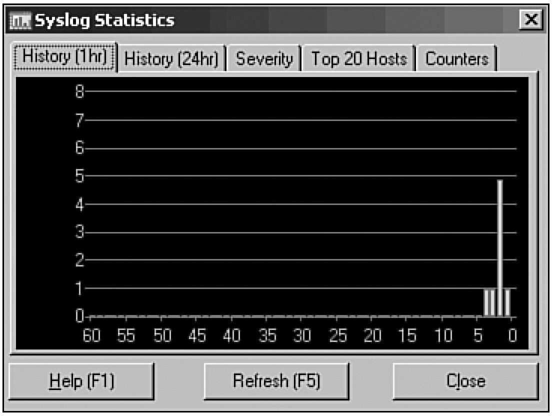
You can also directly open this file in MS-Excel to sort the messages. Moreover, you can choose other file formats for the log files by selecting options on the **Log file format** drop-down menu in the Kiwi Syslog Daemon Setup window. To navigate to the choices (from the Kiwi Syslog Service Manager window), choose **File > Setup** to launch the Kiwi Syslog Daemon Setup window. From within this Setup window, navigate to **Rules > Default > Actions > Log to file > Log File Format**.

To view a graphical summary of syslog statistics, choose **View > View Syslog Statistics** in the Service Manager window**.** Fig 4-11 shows the 1-hour history report for syslog statistics. Each bar represents the number of messages received during each 1-minute interval. The chart scrolls from right to left, and the rightmost bar (0) indicates the current traffic.

Apart from the 1-hour history report, the Syslog Statistics windows also features four other tabs, as shown in Figure 4-11. These tabs are History (24hr), Severity, Top 20 Hosts, and Counters. The History (24hr) tab shows a bar chart of the last 24 hours of traffic and is similar to the 1-hour history window.

The Severity tab lists a summary of messages by priority level. The Top 20 Hosts tab comes in handy for quickly identifying chatty hosts. A large number of messages from a particular host indicates a problem on that device.

Figure 4-11 *Syslog Statistics*



The Counters tab reports the traffic and error statistics for the syslog server. The **Messages - Average** counter is handy for setting maximum thresholds for alarm notification.

NOTE Cisco offers the PIX Firewall Syslog Server (PFSS), an MS-Windows–based syslog server for Cisco PIX Firewalls. It runs as a service, has no GUI, listens on either UDP or TCP ports, and is controlled through the Windows Services management console. It can also be used as a syslog server for other devices. The syslog messages are stored in the following text file: C:\Program Files\Cisco\PIX Firewall Syslog Server\pfss.log. The PFSS can be downloaded by registered users from the Cisco website.

Securing Syslog Servers

So far, you have learned about deploying Linux- and Windows-based syslog servers to collect syslog messages from Cisco devices in a network. However, regardless of the type of operating system or the type of the syslog server, the biggest drawback of the syslog protocol is security. The syslog protocol inherently lacks security for the following reasons:

- **Clear text**—Because a syslog sends its information in clear text, a sniffer on the network can easily capture the messages. To avoid this, syslog messages should be sent on a separate network using a second network interface, if possible. You can also use IP Security (IPSec) tunnels to encrypt the traffic flowing to the syslog server.
- **UDP**—Because syslog uses UDP, an attacker can spoof the source address and send spurious messages to the syslog server. Users should use syslog over TCP, if possible, to mitigate this threat.

- **Centralized location**—While centralized logging is good for Netadmins, it is equally good for attackers. If the central syslog server is compromised, the attacker can delete all the syslog messages to clean up his trail. Netadmins should regularly update the syslog server with the latest service packs and security patches.

Configuring Cisco Devices to Use a Syslog Server

Most Cisco devices use the syslog protocol to manage system logs and alerts. But unlike their PC and server counterparts, Cisco devices lack large internal storage space for storing these logs. To overcome this limitation, Cisco devices offer the following two options:

- **Internal buffer**—The device's operating system allocates a small part of memory buffers to log the most recent messages. The buffer size is limited to few kilobytes. This option is enabled by default. However, when the device reboots, these syslog messages are lost.
- **Syslog**—Use a UNIX-style SYSLOG protocol to send messages to an external device for storing. The storage size does not depend on the router's resources and is limited only by the available disk space on the external syslog server. This option is not enabled by default.

TIP Before configuring a Cisco device to send syslog messages, make sure that it is configured with the right date, time, and time zone. Syslog data would be useless for troubleshooting if it shows the wrong date and time. You should configure all network devices to use NTP. Using NTP ensures a correct and synchronized system clock on all devices within the network. Setting the devices with the accurate time is helpful for event correlation.

To enable syslog functionality in a Cisco network, you must configure the built-in syslog client within the Cisco devices.

Cisco devices use a severity level of warnings through emergencies to generate error messages about software or hardware malfunctions. The debugging level displays the output of debug commands. The Notice level displays interface up or down transitions and system restart messages. The informational level reloads requests and low-process stack messages.

Configuring Cisco Routers for Syslog

To configure a Cisco IOS-based router for sending syslog messages to an external syslog server, follow the steps in Table 4-11 using privileged EXEC mode.

Table 4-11 *Configuring Cisco Routers for Syslog*

Step	Command	Purpose
1	Router# **configure terminal**	Enters global configuration mode.
2	Router(config)# **service timestamps** *type* **datetime** [*msec*] [*localtime*] [*show-timezone*]	Instructs the system to timestamp syslog messages; the options for the *type* keyword are **debug** and **log**.
3	Router(config)#**logging** *host*	Specifies the syslog server by IP address or host name; you can specify multiple servers.
4	Router(config)# **logging trap** *level*	Specifies the kind of messages, by severity level, to be sent to the syslog server. The default is informational and lower. The possible values for *level* are as follows: Emergency: **0** Alert: **1** Critical: **2** Error: **3** Warning: **4** Notice: **5** Informational: **6** Debug: **7** Use the debug level with caution, because it can generate a large amount of syslog traffic in a busy network.
5	Router(config)# **logging facility** *facility-type*	Specifies the facility level used by the syslog messages; the default is **local7**. Possible values are **local0**, **local1**, **local2**, **local3**, **local4**, **local5**, **local6**, and **local7**.
6	Router(config)# **End**	Returns to privileged EXEC mode.
7	Router# **show logging**	Displays logging configuration.

NOTE When a level is specified in the **logging trap** *level* command, the router is configured to send messages with lower severity levels as well. For example, the **logging trap** warning command configures the router to send all messages with the severity warning, error, critical, and emergency. Similarly, the **logging trap** *debug* command causes the router to send all messages to the syslog server. Exercise caution while enabling the debug level. Because the debug process is assigned a high CPU priority, using it in a busy network can cause the router to crash.

Example 4-12 prepares a Cisco router to send syslog messages at facility local3. Also, the router will only send messages with a severity of warning or higher. The syslog server is on a machine with an IP address of 192.168.0.30.

Example 4-12 *Router Configuration for Syslog*

```
Router-Dallas#
Router-Dallas#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router-Dallas(config)#logging 192.168.0.30
Router-Dallas(config)#service timestamps debug datetime localtime show-timezone
  msec
Router-Dallas(config)#service timestamps log datetime localtime show-timezone msec
Router-Dallas(config)#logging facility local3
Router-Dallas(config)#logging trap warning
Router-Dallas(config)#end
Router-Dallas#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 79 messages logged
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: disabled
    Trap logging: level warnings, 80 message lines logged
        Logging to 192.168.0.30, 57 message lines logged
```

Configuring a Cisco Switch for Syslog

To configure a Cisco CatOS-based switch for sending syslog messages to an external syslog server, use the privileged EXEC mode commands shown in Table 4-12.

Table 4-12 *Configuring a Cisco Switch for Syslog*

Step	Command	Purpose
1	Switch>(enable) **set logging timestamp** {**enable** | **disable**}	Configures the system to timestamp messages.
2	Switch>(enable)**set logging server** *ip-address*	Specifies the IP address of the syslog server; a maximum of three servers can be specified.
3	Switch>(enable) **set logging server severity** *server_severity_level*	Limits messages that are logged to the syslog servers by severity level.

Table 4-12 *Configuring a Cisco Switch for Syslog (Continued)*

Step	Command	Purpose
4	Switch>(enable) **set logging server facility** *server_facility_parameter*	Specifies the facility level that would be used in the message. The default is **local7**. Apart from the standard facility names listed in Table 4-1, Cisco Catalyst switches use facility names that are specific to the switch. The following facility levels generate syslog messages with fixed severity levels: **5**: System, Dynamic-Trunking-Protocol, Port-Aggregation-Protocol, Management, Multilayer Switching **4**: CDP, UDLD **2**: Other facilities
5	Switch>(enable) **set logging server enable**	Enables the switch to send syslog messages to the syslog servers.
6	Switch>(enable) **Show logging**	Displays the logging configuration.

Example 4-13 prepares a CatOS-based switch to send syslog messages at facility local4. Also, the switch will only send messages with a severity of warning or higher. The syslog server is on a machine with an IP address of 192.168.0.30.

Example 4-13 *CatOS-Based Switch Configuration for Syslog*

```
Console> (enable) set logging timestamp enable
System logging messages timestamp will be enabled.
Console> (enable) set logging server 192.168.0.30
192.168.0.30 added to System logging server table.
Console> (enable) set logging server facility local4
System logging server facility set to <local4>
Console> (enable) set logging server severity 4
System logging server severity set to <4>
Console> (enable) set logging server enable
System logging messages will be sent to the configured syslog servers.
Console> (enable) show logging
Logging buffered size: 500
timestamp option: enabled
Logging history size: 1
Logging console: enabled
Logging server: enabled
{192.168.0.30}
server facility: LOCAL4
server severity: warnings(4
Current Logging Session: enabled

Facility            Default Severity        Current Session Severity
------------        ----------------------  -----------------------
```

continues

Example 4-13 *CatOS-Based Switch Configuration for Syslog (Continued)*

```
cdp                 3                        4
drip                2                        4
dtp                 5                        4
dvlan               2                        4
earl                2                        4
fddi                2                        4
filesys             2                        4
gvrp                2                        4
ip                  2                        4
kernel              2                        4
mcast               2                        4
mgmt                5                        4
mls                 5                        4
pagp                5                        4
protfilt            2                        4
pruning             2                        4
radius              2                        4
security            2                        4
snmp                2                        4
spantree            2                        4
sys                 5                        4
tac                 2                        4
tcp                 2                        4
telnet              2                        4
tftp                2                        4
udld                4                        4
vmps                2                        4
vtp                 2                        4

0(emergencies)        1(alerts)             2(critical)
3(errors)             4(warnings)           5(notifications)
6(information)        7(debugging)
Console> (enable)
```

Configuring a Cisco PIX Firewall for Syslog

Proactive monitoring of firewall logs is an integral part of a Netadmin's duties. The firewall syslogs are useful for forensics, network troubleshooting, security evaluation, worm and virus attack mitigation, and so on. The configuration steps for enabling syslog messaging on a PIX are conceptually similar to those for IOS- or CatOS-based devices. To configure a Cisco PIX Firewall with PIX OS 4.4 and above, perform the steps shown in Table 4-13 in privileged EXEC mode.

Table 4-13 *PIX Configuration for Syslog*

Step	Command	Purpose
1	Pixfirewall# **config terminal**	Enters global configuration mode.
2	Pixfirewall(config)#**logging timestamp**	Specifies that each syslog message should have a timestamp value.
3	Pixfirewall(config)#**logging host** [*interface connected to syslog server*] *ip_address* [*protocol*/*port*]	Specifies a syslog server that is to receive the messages sent from the Cisco PIX Firewall. You can use multiple **logging host** commands to specify additional servers that would all receive the syslog messages. The *protocol* is UDP or TCP. However, a server can only be specified to receive either UDP or TCP, not both. A Cisco PIX Firewall only sends TCP syslog messages to the Cisco PIX Firewall syslog server.
4	Pixfirewall(config)#**logging facility** *facility*	Specifies the syslog facility number. Instead of specifying the name, the PIX uses a 2-digit number, as follows: local0 - **16** local1 - **17** local2 - **18** local3 - **19** local4 - **20** local5 - **21** local6 - **22** local7 - **23** The default is **20**.

continues

Table 4-13 *PIX Configuration for Syslog (Continued)*

Step	Command	Purpose
5	pixfirewall(config)#**logging trap** *level*	Specifies the syslog message level as a number or string. The *level* that you specify means that you want that *level* and those values less than that *level*. For example, if *level* is **3**, syslog displays **0**, **1**, **2**, and **3** messages. Possible number and string *level* values are as follows: **0**: Emergency; System-unusable messages **1**: Alert; Take immediate action **2**: Critical; critical condition **3**: Error; error message **4**: Warning; warning message **5**: Notice; normal but significant condition **6**: Informational: information message **7**: Debug; debug messages and log FTP commands and WWW URLs
6	pixfirewall(config)#**logging on**	Starts sending syslog messages to all output locations.
7	pixfirewall(config)#**no logging message** <*message id*>	Specifies a message to be suppressed.
8	pixfirewall(config)#**exit**	Exits global configuration mode.

Example 4-14 prepares the Cisco PIX Firewall to send syslog messages at facility local5 and severity debug and below to the syslog server. The Netadmin does not want the PIX to log message 111005. The syslog server has an IP address of 192.168.0.30.

Example 4-14 *Configuring a Cisco PIX Firewall for Syslog*

```
Firewall-Dallas#
Firewall-Dallas# config terminal
Firewall-Dallas(config)# loggin time
Firewall-Dallas(config)# logging host 192.168.0.30
Firewall-Dallas(config)# logging facility 21
Firewall-Dallas(config)# logging trap 7
Firewall-Dallas(config)# logging on
Firewall-Dallas(config)# no logging message 111005
rewall-Dallas(config)# exit
Firewall-Dallas# show logging
Syslog logging: enabled
    Facility: 21
    Timestamp logging: enabled
    Standby logging: disabled
```

Example 4-14 *Configuring a Cisco PIX Firewall for Syslog (Continued)*

```
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: disabled
    Trap logging: level debugging, 6 messages logged
        Logging to inside 192.168.0.30
    History logging: disabled
    Device ID: disabled
```

For added reliability, the Cisco PIX Firewall can be configured to send syslog messages through TCP. Please note that if the syslog server disk is full, it can close the TCP connection. This will cause a denial of service because the Cisco PIX Firewall will stop all traffic until the syslog server disk space is freed. Both Kiwi Syslogd Server and PFSS offer this feature. Kiwi Syslogd has an alert mechanism to warn the Netadmin through e-mail or pager when the disk is nearing its capacity. The setting can be established from the Syslog Daemon Setup window, as shown in Figure 4-9, for Kiwi syslog configuration.

If the PIX stops because of a disk-full condition, you must first free some disk space. Then disable syslog messaging on the PIX by using the **no logging host** *host* command, followed by reenabling syslog messaging using the **logging host** *host* command.

CAUTION The change in facility level for a particular message in the previous example is for illustration purposes only. Changing the facility level from its default value is an advanced Netadmin function and is strongly discouraged.

Example 4-15 shows the configuration steps for a Cisco PIX Firewall to send syslog messages at TCP port 1468.

Example 4-15 *PIX Configuration for TCP Syslog*

```
Firewall-Dallas# config terminal
Firewall-Dallas(config)# logging host inside 192.168.0.30  tcp/1468
Firewall-Dallas(config)# exit
Firewall-Dallas# show logging
Syslog logging: enabled
    Facility: 21
    Timestamp logging: enabled
    Standby logging: disabled
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: disabled
    Trap logging: level debugging, 12 messages logged
        Logging to inside 192.168.0.30 tcp/1468
    History logging: disabled
    Device ID: disabled
Firewall-Dallas#
```

CAUTION A Cisco PIX Firewall facing the Internet is subjected to a large amount of unsolicited traffic in the form of ping scans, port scans, and probes. This can cause the log file to become large within days. It will be filled with data, making it difficult to search for useful information. You should fine-tune your firewall to suppress certain common messages using the **no logging message** *message-id-number* command. Additionally, use the IOS firewall features on the edge router to filter unwanted traffic before it hits the Cisco PIX Firewall.

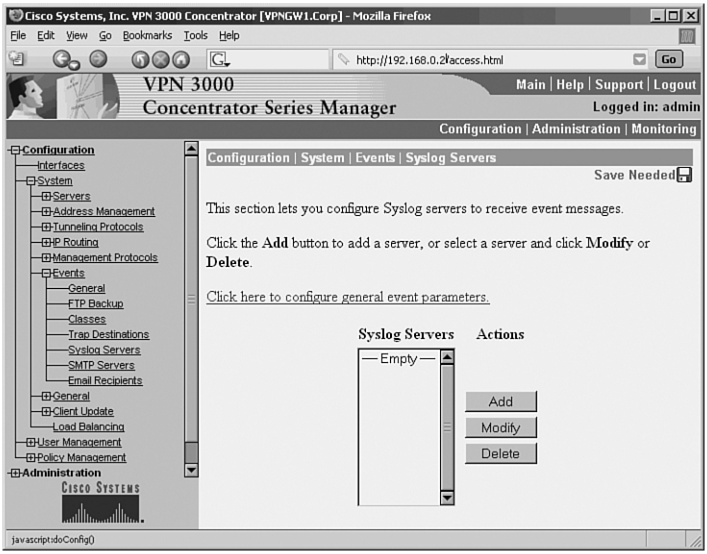
Configuring a Cisco VPN Concentrator for Syslog

The Cisco VPN 3000 Series Concentrator provides an appliance-based solution for deploying VPN functionality across remote networks. VPN concentrators are often connected parallel to the firewalls, as shown earlier in Figure 4-1. The design simplifies the management of the network but creates security concerns. After a user has been authenticated through VPN concentrators, the user has complete access to the network. This makes a strong case for logging the messages from the VPN concentrator. To configure the Cisco VPN 3000 Series Concentrator for sending syslog messages, follow these steps:

Step 1 Log in to the VPN concentrator using a web browser.

Step 2 Navigate to the syslog server page by choosing **Configuration > System > Events > Syslog Servers**, as shown in Figure 4-12.

Figure 4-12 *VPN Concentrator—Syslog Server*



This section lets you configure Syslog servers to receive event messages.

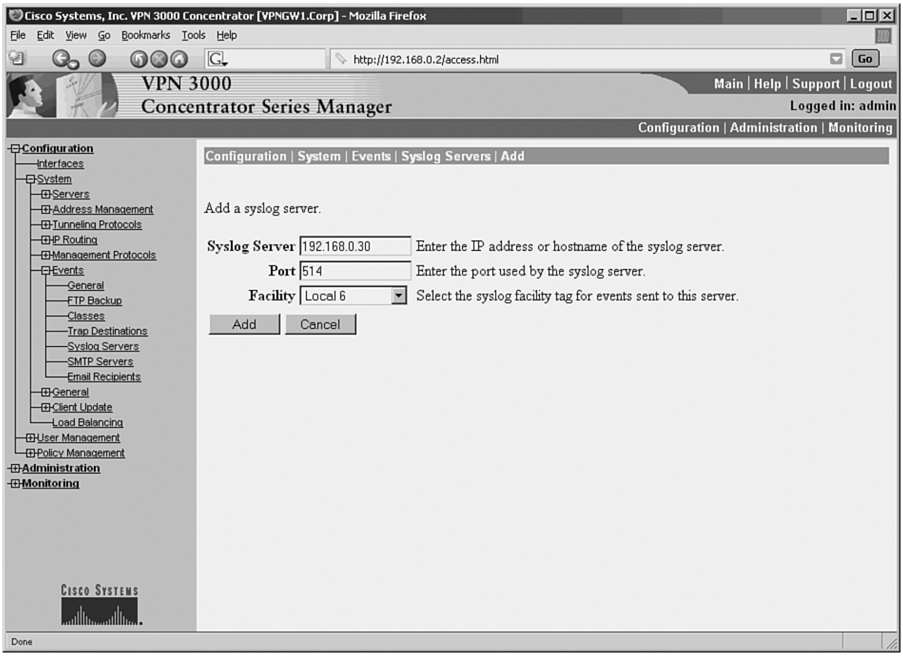
Click the **Add** button to add a server, or select a server and click **Modify** or **Delete**.

Click here to configure general event parameters.

Step 3 On the Syslog Servers page, click the **Add** button (see Figure 4-12).

Step 4 Enter the IP address of the syslog server and select the facility level from the Facility drop-down menu, as shown in Figure 4-13. Save these settings and return to the Syslog Servers page by clicking the **Add** button.

Figure 4-13 *VPN Concentrator—Add Syslog Server*



Add a syslog server.

Syslog Server 192.168.0.30 Enter the IP address or hostname of the syslog server.

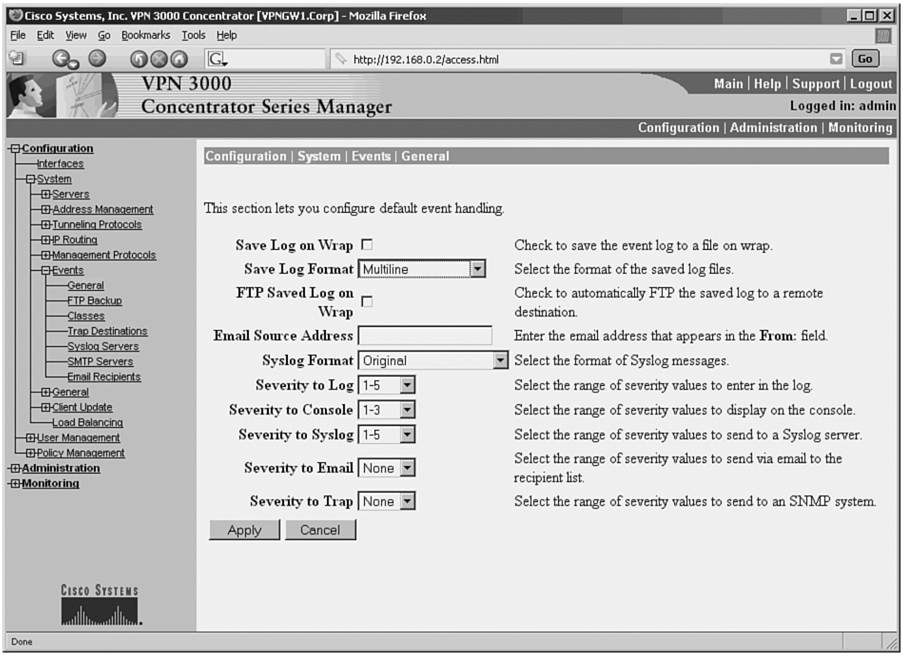
Port 514 Enter the port used by the syslog server.

Facility Local 6 Select the syslog facility tag for events sent to this server.

Step 5 To select the kind of messages that are to be sent to the syslog server, navigate to the General page by choosing **Configuration > System > Events > General**.

Step 6 On the General page, select an option from the Severity to Syslog drop-down menu, as shown in Figure 4-14, and click the **Apply** button.

Figure 4-14 *VPN Concentrator—General Configuration*



Configuration | System | Events | General

This section lets you configure default event handling.

Save Log on Wrap	□	Check to save the event log to a file on wrap.
Save Log Format	Multiline	Select the format of the saved log files.
FTP Saved Log on Wrap	□	Check to automatically FTP the saved log to a remote destination.
Email Source Address		Enter the email address that appears in the **From:** field.
Syslog Format	Original	Select the format of Syslog messages.
Severity to Log	1-5	Select the range of severity values to enter in the log.
Severity to Console	1-3	Select the range of severity values to display on the console.
Severity to Syslog	1-5	Select the range of severity values to send to a Syslog server.
Severity to Email	None	Select the range of severity values to send via email to the recipient list.
Severity to Trap	None	Select the range of severity values to send to an SNMP system.

Apply Cancel

Step 7 To save the configuration changes, click the **Save Needed** icon.

As configured in this example, the VPN concentrator is now ready to send syslog messages at facility local6, severity 1–5 to server 192.168.0.30.

Commercial Cisco Products

In addition to the free PFSS, Cisco also offers a syslog server that is integrated into the CiscoWorks suite of network-management products. The syslog server with an advanced reporting engine is part of the Resource Manager Essentials (RME) module of CiscoWorks. The RME offers the following features:

- Provides web-based applications
- Maintains a database of current network information
- Generates a variety of reports that can be used for troubleshooting and capacity planning
- Periodically retrieves and updates device information, such as hardware, software, and configuration files

- Automatically records changes made to network devices, making it easy to identify when changes are made and by whom
- Deploys Cisco software images and views configurations of Cisco routers and switches
- Links to Cisco.com service and support for Terminal Access Controller (TAC) case management

Because RME has advanced features integrated with the syslog server, it can correlate the syslog events with other functions and provide enhanced reporting.

Summary

The topics covered in this chapter prepare the Netadmin for deploying a centralized logging facility to collect syslog messages from all the network devices. The Netadmins should be able to perform the following syslog-related tasks:

- Deploy a Linux-based syslog server to support Cisco devices
- Deploy a Microsoft Windows–based syslog server
- Centrally log events and alarms generated by Cisco IOS–based routers and switches
- Centrally log events and alarms generated by Cisco CatOS–based switches
- Centrally log events and alarms generated by Cisco PIX Firewalls
- Centrally log events and alarms generated by Cisco VPN 3000 Series Concentrators.

Table 4-14 provides a list of all the tools discussed in this chapter. The table also provides the source of documentation for each tool.

Table 4-14 *Tools Discussed in Chapter 4*

Tool	Function	Supported OS	Installable Files	Documentation Sources
Syslogd	UNIX-style syslog	Part of the standard Linux/UNIX OS for handling system and kernel logs	Part of the standard OS; installation not required	man syslogd
Syslog-ng	UNIX-style syslog with advanced filtering capabilities	Linux/UNIX	http://www.balabit.com	http://www.campin.net/var/docs/syslog-ng/html http://www.campin.net/usr/share/doc/syslog-ng/html/book1.html INSTALL and README docs included in the source code

continues

Table 4-14 *Tools Discussed in Chapter 4 (Continued)*

Tool	Function	Supported OS	Installable Files	Documentation Sources
Kiwi Syslogd Server	UNIX-style syslog with e-mails and alerts; also works for Cisco PIX Firewall syslogging	Windows 95, 98, Me, NT 4, 2000, 2003, XP	http://www.kiwisyslog.com	Help file with system management window
PIX Firewall syslog server	UNIX-style syslog; has no GUI; works for Cisco PIX Firewalls as well as other devices	Windows NT4 with SP6, 2000, XP	http://www.cisco.com (user needs cisco.com login)	Documentation file readme.rtf installed with the server
3Com syslog server	UNIX-style syslog	Windows 95/98/NT	http://support.3com.com/software/utilities_for_windows_32_bit.htm	Help file with the GUI