Lire User's Manual

Joost van Baal

Wessel Dankers

Francis J. Lacoste

Wolfgang Sourdeau

Egon L. Willighagen

Lire User's Manual

by Joost van Baal, Wessel Dankers, Francis J. Lacoste, Wolfgang Sourdeau, and Egon L. Willighagen

Revision History

Revision 2.1.1 $Date: 2008/07/17 13:47:14 $
$Id: user-manual.dbx,v 1.92 2008/07/17 13:47:14 wraay Exp $

Table of Contents

List of Examples

Preface

Log file analysis is both an essential and tedious part of system administration. It is essential because it's the best way of profiling the usage of the service installed on the network. It's tedious because programs generate a lot of data and tools to report on this data are unavailable or incomplete and when such tools exists, they are specific to one product, which means that you can't compare your qmail and Exim mail servers.

Lire is a software package developed by the Stichting LogReport Foundation to generate useful reports from raw log files of various network programs. Multiple programs are supported for various types of network services. Lire also supports various output formats for the generated reports.

What This Book Contains

This book is the *Lire User's Manual*. It describes how to install, configure and use Lire. The intended audience is system administrators who want to install and use Lire to gather informations about the services operating on their network.

There is another book, the *Lire Developer's Manual* that is intended for system administrators or programmers that want to extend Lire or want to understand its architecture and design.

How Is This Book Organized?

This book is divided into three parts. Part I gives an overview of what Lire can achieve for you. It explains how to install Lire and gives simple usage patterns for various kinds of environments.

Part II describes most the supported log files Lire.

Finally, you will find in Part III reference material on all installation options and on all the runtime parameters of Lire.

If You Don't Find Something In This Manual

You can report typos, incorrect grammar or any other editorial problem to <`bugs@logreport.org`>. We welcome reader's feedback. If you feel that certain parts of this manual aren't clear, are missing information or lacking in any other aspect, please tell us. Of course, if you feel like writing the missing information yourself, we'll very happily accept your patch. We will make our best effort to improve this manual.

Remember, that there is another manual, the *Lire Developer's Manual* which contains comprehensive information on how to extend Lire and describes in detail its internal architecture and design.

There are various mailing lists for Lire's users. There is a general users' discussion list where you can find help on how to install and use Lire. You can subscribe to this mailing list by sending an empty email with a subject of *subscribe* to <`questions-request@lists.logreport.org`>. Email for the list should be sent to <`questions@lists.logreport.org`>.

You can keep track of Lire's new release by subscribing to the announcement mailing list. You can subscribe yourself by sending an empty email with a subject of *subscribe* to <announcement-request@lists.logreport.org>.

Finally, if you're interested in Lire's development, there is a development mailing list to which you can subscribe by sending an empty email with a subject of *subscribe* to <development-request@lists.logreport.org>. Email to the list should be sent to <development@lists.logreport.org>.

I. Lire Overview

Chapter 1. Introducing Lire

What Is Lire?

The Lire package is targeted at automatically generating useful reports from raw log files from various services. Currently, Lire can generate reports for a variety of email, web, dns, ftp, print servers and firewalls, and supports multiple output formats. Lire is developed by the Stichting LogReport Foundation, more information about the project can be found on http://www.logreport.org/.

Lire is built around the concept of a *superservice*. A superservice is a class of applications which share the same reports. Lire supports many superservices like dns, email, firewall, ftp, print and www. This means that log files for all supported email servers (*service* in Lire's parlance) will get similar reports. This is important for heterogeneous environments where you could have e.g. Sendmail and Postfix mail servers running. You will get similar reports which you can compare.

Lire is also very modular. Most of its features are based on plugins. For example, each output formats and log formats are supported via a plugin. If you can program perl, it is very easy to develop a plugin. You should consult the *Lire Developer's Manual* for all the information.

Lire can be used as a log database system (called a *DLF Store*). In this setup you configure periodical importation of log data and periodical generation of reports from that log data.

There are also commands that can be use to generate, mege and formats reports on an on an ad-hoc basis.

The Lire distribution also includes a bunch of shell scripts which enable to set up an online responder system. In this setup, the Lire system receives emails containing log files from other hosts and sends generated reports back by email. Optionally, the log files can be anonymized before being sent.

Supported Systems

The package is reported to be useable on

- GNU/Linux (Debian GNU/Linux, Red Hat Linux, Mandrake Linux and probably a variety of other distributions)
- BSD (FreeBSD, OpenBSD, NetBSD, Mac OS X)
- Solaris
- It should run on any modern UNIX with a recent version of perl.

The LogReport team generally tests Lire on various GNU/Linux distributions, as well as on OpenBSD before shipping. Don't worry if your system isn't listed here: it means we haven't had the opportunity to test Lire on your system, it does *not* mean Lire won't run on your system. If Perl runs on your system (which very likely is the case), Lire very likely will run on it too. However, please send us a note on your experiences. We're interested in Lire's portability.

Supported Applications

Lire can generate reports for a variety of dns, email, print, proxy, database, ftp and web servers as well as some firewalls. You can find the definitive list of supported log formats by running the command **`lr_log2report --help dlf-converters`**.

Database

Lire can generate reports from the log files of database servers:

- MySQL. http://www.mysql.org/

For these applications, you will get reports about the number of queries, the top users, the most used databases and more.

Dialup

Lire can generate reports from the log files of Linux kernel 2.4.x isdnlog log files:

- Linux kernel 2.4.x isdnlog http://www.isdn4linux.de/

DNS

Lire can generate reports from the query log files of two DNS servers:

- Bind 8. http://www.isc.org/products/BIND/bind8.html
- Bind 9. http://www.isc.org/products/BIND/bind9.html

For these applications, you will get reports about the number of DNS requests by hour, the top DNS clients, the most requested names and more.

DNS Zone

Lire can generated reports from DNS server logs about DNS Zone transfers: AXFR's and the loading of zones, as logged by e.g BIND 8's named log.

Email

Six email servers are supported by Lire:

- ArGoSoft Mail Server. http://www.argosoft.com/applications/mailserver/ (http://www.argosoft.org/)
- Exim. http://www.exim.org/
- Postfix. http://www.postfix.org/
- Netscape Messaging Server.
- Qmail. http://www.qmail.org/
- Sendmail. http://www.sendmail.org/

The email servers' reports will show you the number of deliveries and the volume of email delivered by day, the domains from which you receive or send the most emails, the relays most used, etc.

Firewall

Several packet filtering firewalls are supported by Lire:

- Log files from Cisco IOS http://www.cisco.com/univercd/cc/td/doc/product/software/ (http://www.cisco.com/univercd/cc/td/doc/product/software/).
- IPfilter log files http://coombs.anu.edu.au/~avalon/ip-filter.html (http://coombs.anu.edu.au/~avalon/ip-filter.html).
- Linux 2.2.X ipchains log files. http://netfilter.samba.org/ipchains/ (http://netfilter.samba.org/ipchains/).
- Linux 2.4.X iptables log files. http://netfilter.samba.org/ (http://netfilter.samba.org/).
- All log files using the WebTrends Enhanced Log Format (http://www.webtrends.com/partners/welfOverview.htm). This makes Lire support a potentially large number of firewall products. Consult http://www.webtrends.com/partners/firewall.htm for a list. Note that we didn't test Lire with all of those products. We appreciate all feedback regarding how Lire behaves with those products.

The reports generated will include informations about the IP address with the largest volume of data denied, the denied TCP ports, etc.

FTP

Lire can generate reports for FTP servers that use the xferlog log format. Some of the FTP servers known to support that log format:

- BSD ftpd. (As found on OpenBSD, FreeBSD and most UNIXes).
- ProFTPD. http://www.proftpd.org/
- Wu-Ftpd. http://www.wu-ftpd.org/

It also supports log files from Microsoft Internet Information Server, which uses a variant of the W3C Extended Log Format.

The ftp superservice reports will include information such as the clients with the most transfers, the most requested files, the most active users, the amount of bytes transferred by day, etc.

Message Store

Lire can generate reports from log files from two message stores:

- Netscape Messaging Server.
- Netscape Messaging Server Mail Multi Plexor

Print

Lire can generate reports for two print servers:

- CUPS http://www.cups.org/
- LPRng http://www.lprng.com/

The reports generated will include information about the usage of the printers, statistics on the jobs and users.

Proxy

Lire supports three types of log files for proxy servers:

- Squid. http://www.squid-cache.org/
- Microsoft Internet Security and Acceleration Server. http://www.microsoft.com/isaserver/
- All log files using the WebTrends Enhanced Log Format (http://www.webtrends.com/partners/welfOverview.htm). This makes Lire support a potentially large number of proxy products. Consult http://www.webtrends.com/partners/firewall.htm for a list. Note that we didn't test Lire with all of those products. We appreciate all feedback regarding how Lire behaves with those products.

Syslog

Lire can generate overview reports about your syslog log files. It supports more than 8 different syslog log file formats.

WWW

Lire supports the three most common log formats for web servers: common log format (CLF), combined log format and the W3C extended log format (http://www.w3.org/TR/WD-logfile.html). Most web servers are able to log in one of those formats. It has been verified that Lire is able to generate reports for the following web servers:

- Apache. http://httpd.apache.org/
- Boa. http://www.boa.org/
- Microsoft Internet Information Server (3.X, 4.X, 5.X).
- iPlanet Web Server.http://www.iplanet.com/

Reports for the www superservice will include information like the number of requests by day, requests by browser, attack detection, top referers, etc. It is Lire's most complete report.

Supported Output Format

Lire supports multiple report output formats. All reports are generated in a native XML format which can be transformed into different other output formats. To find the available output formats on your system, you can run **`lr_log2report --help output-formats`**. The following formats are supported:

Text

The default output format is simple text. Simple text reports are best used for daily email reports.

(X)HTML

Lire can generate HTML reports that can be viewed in any web browser. Those reports can include charts for easy overview. The HTML output formats can be generated using multiple files or all in one file.

PDF

To print the reports, Lire can generate Adobe PDF output or PostScript. Like the HTML reports, those can include charts for easy overview. The intermediary formats (LaTeX and DVI) used for these output foramts are also available output formats.

Excel 95

The reports can be formatted as an Excel95 spreadsheet that can be read by OpenOffice, KSpread, Gnumeric and many other spreadsheets applications.

What Lire Can't Do

Even with all the reports available, all those applications supported and all the possible output formats, there are still a number of things that Lire can't do by design. Lire is a *batch report generator*, it isn't a *real-time log analyzer*. There are a lot of real-time alerting tools out there. Lire is designed to generate reports from log files periodically (usually after the log files are rotated).

In case you find something you would like to see Lire do and it is reasonable that Lire should be able to do it, please let us know. In the Section called *If You Don't Find Something In This Manual* in *Preface* you can find how to get in contact with us.

Chapter 2. Installing Lire

Lire supports various installation environments. This chapter contains all there is to know about the installation of Lire in various setup scenarios: from the simple client setup to the installation of an online responder. You can find some quick installation instructions in the `INSTALL` file. Please note: if your vendor ships a prepackaged Lire system, you're likely better of using that one. Furthermore, Stichting LogReport Foundation offers prebuild packages like RPM's. Be sure to look around for these before deciding to do a manual installation using the Lire tarball.

Client Installation

The simplest setup to install Lire in a client-server scenario is where the log files are sent by email to an online responder for processing.

> **Tip:** You can test Lire by using Stichting LogReport Foundation's online responder available at <`log@service.logreport.org`>. (To process sendmail log files, send them to <`log@sendmail.logreport.org`>).

Requirements

To use Lire in such a setup, you only need a mailer (any will do) and an email address where the generated report can get sent to.

Installing

No special installation is necessary. You can generate reports by sending the log files to the responder right away. Consult the the Section called *Using A Responder* in Chapter 3 for the complete story.

Standalone Installation

The most common installation scenario will be where you install Lire on one system to generate daily or weekly reports from cron or by using the command line tools. This setup will install the complete software.

Requirements

Minimum Requirements

To install Lire on a system, you need the following:

- GNU gzip.
- Perl 5.6.1 or later (5.8.3 strongly recommended).
- The XML::Parser perl module. (This one needs the expat library.)

  XML::Parser is available from any CPAN mirror. (http://www.cpan.org/modules/by-module/XML/).

  The expat library is available from http://expat.sourceforge.net/.
- The DBD::SQLite perl module, available from http://www.cpan.org/modules/by-module/DBD/ (which in turn requires DBI from http://www.cpan.org/modules/by-module/DBI/).
- The libintl-perl perl module, available from http://www.cpan.org/modules/by-module/Locale/.
- The Curses::UI perl module, available from ftp://ftp.cpan.org/pub/CPAN/modules/by-module/Curses/. Curses is required as well and is available from the same location.
- Standard UNIX utilities like **sh**, **ls**, **grep**, **bc**, **cut**, **head**, **sort**, **tar**, etc.

Those are the minimal requirements. With those, you will be able to generate text, html and latex reports.

Requirements for Other Output Formats

Some output formats have other requirements:

- The method to render charts is through the use of **ploticus**. The **ploticus** generates nice looking graphs, especially in combination with HTML, PostScript or PDF output. As a standalone program it is quite easy to install (depending on the operating system you use).

  The **ploticus** program is available from http://ploticus.sourceforge.net/). This package contains everything necessary to render GIF, SVG and PostScript images. The site offers binaries for various platforms.
- To generate DVI or PostScript reports you will need a recent TeX installation which contains Omega (**lambda** and **odvips**). The popular teTeX distribution ships Omega since version 0.9.

  The teTeX distribution is available from http://www.tug.org/teTeX/.
- To generate PDF reports, in addition to a recent TeX installation, you will need the command **ps2pdf** which comes with the GhostScript PostScript interpreter.

  The GhostScript PostScript interpreter is available from http://www.cs.wisc.edu/~ghost/.
- To generate Excel95 reports you will need the Spreadsheet::WriteExcel perl module, available from http://www.cpan.org/modules/by-module/Spreadsheet/ (which in turn requires Parse::RecDescent from http://www.cpan.org/modules/by-module/Parse/).

Other Optional Requirements

Other optional things you may want to install:

- When available, the **logger** utility can be used to send Lire output to syslog.
- The Time-modules perl module (available from any CPAN mirror, http://www.cpan.org/modules/by-module/Time/. If it isn't present in the system, the required files included with Lire will be installed.
- The MIME-Tools perl module (available from any CPAN mirror, http://www.cpan.org/modules/by-module/MIME/.

  This module is necessary to conveniently send reports by email or to operate a responder.

Installing

Installation of Lire is pretty straightforward:

1. Make sure that you have the requirements installed.
2. Extract the source code:

   ```
   $ gzip -dc lire-version.tar.gz | tar xf -
   ```

3. Configure the software. You may use the `--prefix` option to specify where you want to install Lire. By default, it will be installed under `/usr/local`.

   ```
   $ cd lire-version
   $ ./configure [--prefix=path]
   ```

   Make sure not to use ~ in the *path*. This is known to fail.

   It find all requirements you had installed.
4. Compile the software, this will consist only of generating man pages).

   ```
   $ make
   ```

5. You may have to become root if you are installing in a directory where only root has write permissions.
6. Install Lire.

   ```
   # make install
   ```

That's it! You have a complete Lire installation and are ready to generate some reports. See Chapter 3 for information on using Lire.

Anonymized Client Installation

Although the client-only setup is the easiest to install and use, some people might understandably be worried about sending log files that may contain sensitive data to a public online responder. That is why Lire supports anonymizing of log files. In an anonymized client setup, hostnames, emails and IP addresses in the log files are anonymized before being sent to the responder. The responder replies with a report in the Lire XML report format which is then de-anonymized by the client and transformed into the appropriate output format.

Requirements

The anonymized client installation has the same requirements as a standalone installation (see the Section called *Standalone Installation*). Like in the Standalone Installation, those will vary according to the output format you want to support.

Additionally, to support the anonymizing process, you will need Berkeley DB and the DB_File perl module. This module is part of the standard perl installation, but on proprietary UNIX systems you may have to install it separately.

Installing

There is no difference between the anonymized client installation and the Standalone Installation procedure. Consult the Section called *Standalone Installation*.

Responder Installation

When you want to generate reports for several servers, it is best to install Lire as a responder on one system to which to other systems can send their log files. This section describes how to setup Lire as a responder.

Requirements

Responder installation has the same requirements as the standalone installation (see the Section called *Standalone Installation*).

There is the following additional requirement:

- The MIME-Tools perl module (available from any CPAN mirror, http://www.cpan.org/modules/by-module/MIME/.

Installation

Basic installation procedure is the same as a standalone installation (see the Section called *Standalone Installation*). You might want to change the `--with-spooldir` option to **configure** (the default is *prefix*`/var/spool/lire`:

```
$ ./configure [--prefix=path --with-spooldir=path_to_spooldir]
```

Lire in a responder setup runs the **lr_spoold** daemon which scans maildirs where requests are delivered. Consequently, to finish the responder installation you have to create a maildir for each service you want to support and setup delivery to those maildirs.

> **Note:** A *maildir* is a mailbox format first developed as part of Qmail where messages are stored in a directory hierarchy instead of a single file. You can find more informations about the maildir format at http://www.courier-mta.org/maildirmake.html.
>
> As far as Lire is concerned, a maildir is a subdirectory *service*`/Maildir/new` which contains email messages in separate files.

The *sysconfdir*`/lire/address.cf` contains the name of the maildirs that are to be scanned and the type of log files that the emails should contain.

Refer to your MTA's documentation for notes on how to setup delivery to maildir. We give some notes on how to do this in the following sections.

The **lr_setup_responder** script can be used to setup some required infrastructure for the responder. Alternatively, one can execute the setup manually: One can create the maildirs by doing e.g.

```
$ cd ~/lire
$ mkdir -p var/spool/lire/common
$ maildirmake var/spool/lire/common/Maildir
$ cd ~/lire/var/spool/lire
$ mkdir bind8_query postfix qmail sendmail
$ maildirmake bind8_query/Maildir
$ maildirmake postfix/Maildir
$ maildirmake qmail/Maildir
$ maildirmake sendmail/Maildir
```

maildirmake gets distributed with qmail and with the Courier Mail Server http://www.courier-mta.org. If you haven't set up delivery to maildirs yet, doing a

```
$ maildirmake foo
```

is about the same as doing

```
$ mkdir foo
$ mkdir foo/cur foo/new foo/tmp
$ chmod og-rwx foo foo/*
```

Installing Under MTA's using procmail as their MDA

On many systems, procmail is used as the default Mail Delivery Agent. For instance, sendmail very often is configured to use procmail. If your MTA is configured like this, you can use procmail to take care of delivering to the right Maildir. We give some hints on how to get this done.

In Lire's `$HOME/.procmailrc` you can put

```
:0:
* ^To:.*combined-log@
<LR_SPOOLDIR>/combined/Maildir/new

:0:
* ^To:.*sendmail-log@
<LR_SPOOLDIR>/sendmail/Maildir/new
```

etc. Make sure to replace *<LR_SPOOLDIR>* by the appopriate path. After that, you'll only have to make sure that the addresses combined-log, sendmail-log, etc. are aliases for the Lire user. You can then run lr_spoold to monitor the spool archives.

Installing Under Exim

There is more than one way to setup maildir delivery on a system running exim http://www.exim.org/. We show only one.

Be sure to have "maildir_format" enabled in the address_directory: section, e.g.

```
address_directory:
  driver = appendfile
  no_from_hack
  prefix = ""
  suffix = ""
  maildir_format
```

in your exim.conf's transport configuration. Furthermore, have "directory_transport" transport in the userforward driver set to "address_directory", e.g.

```
userforward:
  driver = forwardfile
  file_transport = address_file
  pipe_transport = address_pipe
```

```
  reply_transport = address_reply
  directory_transport = address_directory
  no_verify
  check_ancestor
  check_local_user
  file = .forward
  modemask = 002
  filter
```

in your `exim.conf`'s directors configuration. Create a maildir, e.g. `~/.lire/var/spool/combined/`. (See the qmail section for how to do this.) Finally, do e.g.

```
$  cat <<EOT > .forward
> # Exim filter
> save \$home/.lire/var/spool/combined/
> EOT
```

One could create more than one maildir, and configure the useraccount to store email messages for different services in different maildirs. We wont go into this such detail here though.

Installing Under qmail

Suppose your configure-time prefix was $HOME/lire.

```
$ cd ~/lire/var/spool/lire/postfix
$ maildirmake Maildir
$ echo './lire/var/spool/lire/postfix/Maildir/' > .qmail-postfix
```

Get mail to postfix@yourhost delivered to hibou-postfix@yourhost, and controlled by `~hibou/.qmail-postfix`:

```
$ su
# cd /var/qmail/control/users
# vi assign
=postfix:hibou:1028:1028:/home/hibou:-:postfix:
```

Get mail to anybody@postfix.yourdomain delivered to the local postfix mailbox:

```
# vi virtualdomains
postfix.yourdomain:postfix
```

Now send your qmail-send process a SIGHUP.

Chapter 3. Running Lire

This chapter describes the various ways that you can use Lire to process log files to generate reports. The next chapter (Chapter 4) explains how you can set up your system to process your log files automatically at regular interval.

Lire's configuration system

Lire holds its configuration in XML files. To change Lire settings, simply fire up **lire** from a terminal and select the Lire⟶Preferences menu. From there you can change various settings. To save your customized configuration file, press the OK button. Otherwise your changes will be saved in the file *HOME*`/.lire/config.xml`. These settings will only affect your account.

From the Preferences screen, you can change the path to all the external programs that Lire used. This is useful if you installed external programs that weren't present when you built and installed Lire. It is also the place to tweak various settings affecting how the reports are generated.

The list and purpose of each setting is described in the help window while using the tool. The **lire** command is also used to manage DlfStores, described more fully in the chapter (Chapter 4).

Using A Responder

The easiest way to generate a report from your log file is to send your log file to a responder. The report will be sent to you by email to the address specified in the *`Reply-To:`* or *`From:`* header. To use a responder, you only need your standard mailer.

To save bandwidth, responders accept log files compressed using **gzip**, **compress** or **zip**. The log file can be sent in the email body or in a MIME attachment.

Note: Although any mailer will do, you should take care of the following when sending your log file:

- Make sure that your mailer won't insert new lines to wrap long log lines.
- Make sure that your mailer sets the standard MIME headers when using transfer encoding.
- When sending the log file as a MIME attachment, make sure that there are no other attachments (such as a signature) after the log file.

As a public service Stichting LogReport Foundation offers an online responder. To use it, you just send your log file to the appropriate responder for the log format you are using. The email addresses available can be found at http://logreport.org/lire/or/ (http://logreport.org/lire/or/).

Example 3-1. Sending a Log File For Processing To A Responder

In this example, a bind8 query log file is sent to the LogReport responder for processing. The report will be sent back to the user who ran the **mail** command.

```
$  mail -s "Bind8 Log" log@bind8-query.logreport.org < \
 /var/log/query.log
```

To save bandwidth, please send big log files in compressed format only. E.g., do:

```
$  mutt -s "`hostname` `date`" -a \
 /var/log/apache/common.log.1.gz log@common.logreport.org < \
 /dev/null
```

For more privacy, it is possible to send an anonymized log to the responder. Consult the Section called *Sending Anonymized Log Files To A Responder* for more information.

Generating A Report From A Log File

To generate a report from a log file, you use the **lr_log2report** command. The usage of this command is

```
$  lr_log2report dlf_converter logfile report
```

The first parameter is the Dlf Converter to use to handle the log file. There is one Dlf Converter by supported log format. To obtain the list of available convertesr, use

```
$  lr_log2report --help dlf-converters
```

See the lr_log2report(1) man page for all the command details or simply issue **`lr_log2report --help`**.

Example 3-2. Generating a Report With lr_log2report

This is the way to generate a report in the text output format for a log file taken from an Apache log server.

```
$ lr_log2report combined /var/log/apache/access_log ~/report.txt
```

Selecting Output Format

Another output format than the default one (usually text) can be selected by using the `--ouput` switch with the **lr_log2report** command.

Example 3-3. Generating A HTML Report

To generate a HTML report from the same log file as above, you would use the following command:

```
$ lr_log2report --output html combined/var/log/apache/access_log apache_report
```

This will create a `apache_report` containing the report. If the **pl** (that is the Ploticus command) is available, the HTML report will contain nice charts.

You can list the output-formats available on your system by running **`lr_xml2report --help output-formats`**.

Merging Reports

Lire supports the merging of reports: one can combine two reports into one bigger report. This can be used to generate e.g. a weekly report from 7 daily reports, or generate a site-wide report from reports about the behaviour of each server on a site.

We describe how to manually merge reports using the command line tools **lr_xml2report**, but the simplest way to use merging is through the DLF store interface which is described at Chapter 4.

We give an example.

Example 3-4. Merging Reports

To process two BIND v9 logfiles, and merge the reports, one would run:

```
$ lr_log2report --ouput xml bind9_query /var/log/named.2.gz \
    $XMLDIR/20020622.xml
$ lr_log2report --output xml bind9_query /var/log/named.1.gz $XMLDIR/20020623.xml
$  lr_xml2report --tempate dns_default \
    --merge $XMLDIR/20020623.xml  $XMLDIR/20020622.xml $ASCIIDIR/20020622-20020623.txt
```

The `--template` parameter is required for merging and specifies the report configuration template that should be used to merge the reports. You should probably use the same than the one that was used when you generate the reports. If you didn't specify one, (like in the above example) you should know that the default template is named *`superservice`*`_default`. The list of available report configuration templates can be displayed by using the **`lr_xml2report --help report-templates`**.

The `--merge` option is used to specify the other XML reports that should be merged before formatting the report. The **lr_xml2mail** command uses the same options for merging.

Gotchas

The merging functionality is very powerful, and allows you to shoot yourself in the foot. We document some pitfalls.

Holes in your reporting period

When merging XML report files xml.3 (2002-06-02 08:50:48 CEST - 2002-06-09 08:05:06 CEST) and xml.1 (2002-06-16 08:18:40 CEST - 2002-06-21 22:13:09 CEST) , the generated report will gladly display "Reporting on period: 2002-06-02 08:50:48 CEST - 2002-06-21 22:13:09 CEST": There is *no* saveguard against forgetting in-between report files.

Take care when changing report configuration template parameters

In some cases, changing the report configuration template just before merging might lead to bogus data in your report.

Consider this case: our firewall template contain a subreport top-pkt-by-src with the `ips_to_show` set to 10, We process some firewall logs, and archive the XML reports. If we change the `ips_to_show` to 100 and merge the XML reports. This could incorrectly omit some IPs! You've got no guarantee the exact top 100 IPs are shown. This is due to the fact the XML reports do *not* contain all information from the log: they're *reports*, after all.

Due to these issues, the merging is implemented with some heuristics: we keep more data than what's requested by the user in the XML report, to be able to handle most after-the-fact merging requests. We've tested the algorithm with a pretty broad range of real-life log files, and found out generally, the merged reports do give a good reflection of what actually has happened on the network: the heuristic is pretty well choosen. However, if you really need guaranteed 100% accurate data, generate your report directly from the raw logs. If you just want a quick overview, the merging is more suitable. Just make sure you're not cranking the limit parameters up too high in this case.

See also the Report Generating chapter in the Lire Architecture part of the *Lire Developer's Manual*.

Sending Anonymized Log Files To A Responder

For more privacy, you can anonymize your log somewhat before sending it to a responder. Lire includes a command called **lr_anonymize** which will transform everything that looks like an IP address, an email or a domain name into an anonymized form (10.0.0.1, 2.0.0.10.in-addr.arpa, 11.example.com, <`john.doe@2.example.com`>, etc.) The mapping between the real value and its anonymized form is saved in a disk database so that you can reverse the process when you receive the report from the responder.

The procedure is quite simple, you just have to filter your log file through **lr_anonymize** and make sure that the subject of your email starts with `anon`.

Example 3-5. Sending An Anonymized Postfix Log File To A Responder

To send an anonymized postfix log file to the Stichting LogReport Foundation responder, you would use a command like:

```
$  grep ' postfix/' /var/log/mail.log | \
    lr_run lr_anonymize /tmp/anon | \
    mail -s "anon Daily Report" log@postfix.logreport.org
```

The `/tmp/anon` is the database that is used to save the mapping between the real and anonymized values.

> **Warning**
>
> **lr_anonymize** will overwrite the content of that database, so if you reuse the database, make sure that you don't have two concurrent requests to a responder because you will lose the first mappings!

Processing The Responder's Results

The responder will generate a report in an XML format specific to Lire. To obtain a "normal" report from this, you first deanonymize it, then run the appropriate converter on the deanonymized report. You use the **lr_xml2report** command to convert a XML report to one of the available output formats.

You can list the output-formats available on your system by running **`lr_xml2report --help output-formats`**.

Example 3-6. Deanonymizing and Generating A HTML Report

To generate a HTML report from the XML report you received from the responder, you would use the following command:

```
$  lr_run lr_deanonymize /tmp/anon < /tmp/anon-report.xml > /tmp/report.xml
$  lr_xml2report --output html /tmp/report.xml /tmp/report.html
```

Running Lire In A Server Cluster

Using Mail

You can monitor a set of maildirs which receive email messages containing log files for the services as listed in `address.cf` by doing something like:

```
$ lr_run lr_spoold
```

This enables you to configure one host as a reporting host (or "online responder"), while other machines send their log files to it by email for processing. (If remote syslogging is used, a cron-driven setup is sufficient.)

A publicly available online responder is running at log@<servicename>.logreport.org; see http://logreport.org/lire/or/ for more information.

Chapter 4. Using DLF Stores

This explains how to use the **lire** and **lr_cron** to set up an automated log database system. The idea is really simple, you set up a database called a DLF Store into which you will periodically import log files. In this store you can configure various reports that will be generated automatically from the data contained in the store. The generated reports are saved in XML format in the Dlf Store and reports over longer period can be generated by aggregating the saved reports data.

The lire user interface

The **lire** uses interactive console interface to configure Lire configuration and Dlf Stores. You navigate around the interface by using the TAB and Meta-TAB keys. On some systems, (notably GNU/Linux and other systems using the ncurses library) you can also use the mouse to navigate the user interface.

Many screens contain an help area which describes the parameter which is currently being edited. You can scroll this area by using the PgUp and PgDown key.

In listboxes, you can use the j or ArrowDown keys to select the item below the current selection. The k or ArrowUp keys can be used to select the item above the current selection. The DEL key will delete the selected item and the INS key will insert a new item in the list. Hitting Ctrl-C will make a copy of the currently selected item. The J (uppercase) and K keys can be used to reorder the items in the list.

> **Note:** It is possible that you encounter messages stating that your screen is too small to use the application. The message also informs you to stop the application and resize the terminal before restarting. *You do not need to stop the application.* If you are using a resizable terminal (like an xterm for example), simply resize the terminal until the messages disappear.

Accessing a Dlf Store

You manage Dlf Stores by running the **lire** command. Open or create a DLF store using the Store⟶Open... or the Store⟶New... menu.

You will then see a list of the Import jobs, the Report jobs, the Report Configurations and the Dlf Streams available in the Dlf Store. configured in this store.

Import Jobs

The list box Import Jobs contains the configured import job that are part of the Dlf Store. An import job schedules importation of a log file in the store.

The description of the parameters needed to setup an import job are described in the help aread.

Report Jobs

The list box Report Jobs contains the reports that are scheduled to be generated from the data in the Dlf Store.

When you edit a Report Job, you'll have to enter a name for the report jobs and add one or more schedules. If might surprise at first that you can add more than one schedule in a report jobs, but the idea is that all the schedules in the same report job can be safely merged together. For example, you might want to generate a daily, weekly and monthly firewall report while only retaining one week of data in the Dlf Store. In that case, the monthly report will automatically be gnerated from the month daily reports.

Warning

The name of a Report Job should contain only alphanumeric alphanumeric characters, hyphens or underscores.

Report Schedules

Each schedule associates the period at which the report is going to be generated. You also select in this screen the the Section called *Report Configurations* that is going to be used to generate the report.

Note: You can select different report configurations for different schedules in the same report jobs. Since the reports can be merged you should take care to select compatible report configurations.

You can also configure one or more output jobs for the schedules.

Ouput Jobs

The report is generated in XML and saved in the Dlf Store. You can configure Output Jobs to format the report in one of the supported output format and either save this report somewhere on the filesystem and/or send it by email to one or more recipients.

Using lr_cron within Cron

To actually execute all that is configured in the Dlf Store, you need to run the **lr_cron** command at regular interval.

Installing cron jobs is really easy since the only parameters given to **lr_cron** are a *period* and a *store*. The lines to add to your crontab should look similar to:

```
0 * * * * /usr/bin/lr_cron hourly /var/lib/lire/my_store
0 1 * * * /usr/bin/lr_cron daily /var/lib/lire/my_store
0 2 * * 0 /usr/bin/lr_cron weekly /var/lib/lire/my_store
0 3 1 * * /usr/bin/lr_cron monthly /var/lib/lire/my_store
0 4 1 1 * /usr/bin/lr_cron yearly /var/lib/lire/my_store
```

You should configure one job for each different period you use. You can also configure it for all period since nothing will be done when there are no scheduled actions to execute.

> **Note:** You should take care to setup the time at which the **lr command** is executed so that it makes sense in regards of your log rotation scheme. For example, if you run the daily schedule at 1AM but you rotate your daily logs at 2AM, lr_cron will act upon the data of two days ago instead of on the data of the previous day.

Report Configurations

The Report Configurations list contains the available report configuration. The items in this list are available for the *`report_config`* parameter when you create the Section called *Report Schedules*.

You create a new report configuration by selecting a template which you later modify to create your report configuration. There is usually one template per superservice. For example, the default template for the www superservice is called `www_default`. There is also the `empty` template that can be used to start with an empty report configuration.

You can edit each report configuration. Each report contains one title and many sections.

In each section, you can configure filters to report on a portion of the DLF data. Only the DLF data from the section's superservice will be filtered. Even if a report section can contain only subreports related to one superservice, it is possible to have sections using different superservices in the same report.

Charts

When you configure the subreport, you can add one or more charts that will be generated from the subreport data when the report is formatted.

DLF Streams

The DLF Streams list box lists the available DLF Steams in the store. A DLF Stream contains all the data related to one DLF Schema. In Lire, a superservice is one DLF Schema, but there are others created by analysers. For example, the www superservice has 7 related schemas. When you select one DLF Stream, you can see on the screen the number of records it contains as well as the starting and ending date of the stream.

You can use the Clean button to remove old DLF records from the store.

The Configure button can be used to configure the stream. From this screen you can select the default amount of days of data to keep in the stream. Use the value `0` to turn on automatic cleaning of the stream.

Also from the same screen, you can select the analyser to use to generate each of the other schemas related to this DLF Stream. Select the `none` analyser to turn off creation of the related streams. If the analyser takes parameters you will be able to tune it by using the ... button.

II. Log Formats

Chapter 5. Database Supported Log Formats

Lire currently only supports the query log of MySQL. This log file contains all the connectinons and queries sent to your database server.

MySQL's Log

The MySQL's log file will contain information about each start and shutdown of your database server, as well as all connections and queries processed by the database server during its session.

Example 5-1. Sample MySQL Log File

```
/usr/sbin/mysqld, Version: 3.23.43-debug-log, started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time                 Id Command    Argument
011226 21:32:57       1 Connect     root@localhost on
011226 21:33:01       1 Query       show tables
011226 21:33:08       1 Query       show databases
011226 21:33:46       1 Quit
011226 21:34:32       2 Connect     Access denied for user: \
  'jdoe@localhost' (Using password: YES)
011226 21:34:42       3 Connect     Access denied for user: \
  'jdoe@localhost' (Using password: YES)
011226 21:35:59       6 Connect     jdoe@localhost on
                     6 Init DB     nmrshiftdb
                     6 Query       SHOW VARIABLES
011226 21:36:00       6 Query       CREATE TABLE molecules \
  (molid INT, CMLcode TEXT)
                     6 Query       CREATE TABLE chemnames \
  (molid INT, autonom TEXT, name TEXT)
```

Chapter 6. Dialup Supported Log Format

Lire supports logs of one dialup connections: Linux kernel 2.4.x isdnlog.

Chapter 7. DNS Supported Log Format

Lire supports query logs of two DNS servers: Bind 8 and Bind 9.

Note: You have to enable query logging in bind, something which is not turned on by default.

Example 7-1. Enabling Query Log In Bind

To enable query logging in Bind 8 or Bind 9, you should add the following to your `named.conf` configuration file:

```
logging {
    channel query_logging {
         file "/var/log/named_querylog"
         versions 3 size 100M;
         print-time yes;                 // timestamp log entries
      };

      category queries {
          query_logging;
      };
};
```

Bind8 Query Log

Bind 8's query logs contain one entry for each DNS query made to the name server. It logs the time of the query (you have to set *`print-time`* to `yes` for this), the IP of the requesting client, the name queried, the type of the query and the protocol. Recursive queries will have a `+` after the `XX` which appears in all query entries.

Example 7-2. Sample Bind 8 Query Log

```
10-Apr-2000 00:01:20.307 XX /10.2.3.4/1.2.3.in-addr.arpa/SOA/IN
10-Apr-2000 00:01:20.308 XX+/10.4.3.2/host.foo.com/A/IN
```

Bind9 Query Log

Bind 9 logs the same information as Bind 8 (except whether the request was recursive or not) but in a different format. Bind 9.3 and later versions support a more elaborate log file format, which includes the recursive/non-recursive request indicator again.

Note: We also support the new date format introduced in Bind9 9.3 which also contains the year (`15-Jul-2002`).

Example 7-3. Sample Bind 9 Query Log

print-severity and *print-category* were set to `yes` to obtain that log. Lire also accepts logs where those are turned off.

```
Feb 25 11:09:43.651 queries: info: client 10.0.0.3#1035: \
    query: 3.example.com.nl IN A -
Feb 25 11:09:48.739 queries: info: client 10.0.0.3#1035: \
    query: 3.example.com.nl IN A -
Feb 25 12:50:32.476 queries: info: client 10.0.0.3#1035: \
    query: 21.example.com.co.uk IN A -
Feb 25 12:50:34.110 queries: info: client 10.0.0.3#1035: \
    query: 22.example.com IN A -
```

Tip: If you are running a pre-9.3 version of Bind 9 and you are missing the recursive flag from Bind 8, it is possible to add back that feature by patching Bind 9. The following patch by by Wytze van der Raay will add a `+` or `-` after the query type to indicate whether the query was recursive or not. Lire will detect that the log file was made by a patched Bind 9.

```
# patch bin/named/query.c to log recursive/non-recursive query indication
SRC=bin/named/query.c
if [ -f ${SRC}.org ]
then
    echo "Patched ${SRC} already in place"
else
    echo "Patch ${SRC} for recursive/non-recursive query indication"
    cp -p ${SRC} ${SRC}.org
    patch -p0 ${SRC} <<\!
--- bin/named/query.c.org       Mon Sep 24 22:57:48 2001
+++ bin/named/query.c   Tue Sep 25 09:55:21 2001
@@ -3272,7 +3272,8 @@
        dns_rdatatype_format(rdataset->type, typename, sizeof(typename));

        ns_client_log(client, NS_LOGCATEGORY_QUERIES, NS_LOGMODULE_QUERY,
-                     level, "query: %s %s %s", namebuf, classname, typename);
+                     level, "query: %s %s %s%s", namebuf, classname, typename,
+                     WANTRECURSION(client) ? "+" : "-");
 }

 void
!
fi
```

Chapter 8. DNS Zone Supported Log Format

Lire supports named log files from BIND 8.

Chapter 9. Email Zone Supported Log Format

Lire supports log files from six different email servers.

ArGoSoft Mail Server

The log files generated by the ArGoSoft Mail Server are supported. For proper operation, you'll need to turn on the following components' logging:

- Log SMTP commands.
- Log SMTP conversations.
- Log to File.

Example 9-1. ArGoSoft Mail Server Log Sample

```
3/17/2002 12:00:03 AM - SMTP connection with 10.0.0.1 [1.example.com] \
    ended. ID=3342
3/17/2002 12:00:22 AM - Requested SMTP connection from 10.0.0.2 \
    [2.example.com]
3/17/2002 12:00:22 AM - (  3345) 220 ArGoSoft Mail Server Pro \
    for WinNT/2000/XP, Version 1.8 (10.0.0.3)
3/17/2002 12:00:23 AM - (  3345) HELO greed
3/17/2002 12:00:23 AM - (  3345) 250 Welcome, 2.example.com \
    [10.0.0.2], pleased to meet you
3/17/2002 12:00:23 AM - (  3345) RSET
3/17/2002 12:00:23 AM - (  3345) 250 Reset state
3/17/2002 12:00:23 AM - (  3345) MAIL FROM:<john.doe.1@1.mail.example.com>
3/17/2002 12:00:23 AM - (  3345) Checking address \
    john.doe.1@1.mail.example.com
3/17/2002 12:00:23 AM - (  3345) Address john.doe.1@1.mail.example.com \
     is local
```

Exim

The standard log file from Exim is supported.

Example 9-2. Exim Log Sample

```
2001-03-27 10:00:11 exim 3.16 daemon started: pid=215, -q30m, \
    listening for SMTP on port 25
2001-03-27 10:00:11 Start queue run: pid=218
2001-03-27 10:00:11 End queue run: pid=218
2001-03-27 10:08:01 Start queue run: pid=736
2001-03-27 10:08:01 End queue run: pid=736
2001-03-27 11:29:10 14hpmo-00002f-00 <= john.doe.25@1.mail.example.com \
    U=root P=local S=757
2001-03-27 11:29:11 14hpmo-00002f-00 => egonw \
    <john.doe.21@1.mail.example.com> D=localuser T=local_delivery
2001-03-27 11:29:11 14hpmo-00002f-00 Completed
```

Netscape Messaging Server

Netscape Messaging Server logs its information with **syslog**. No special configuration is necessary.

Example 9-3. Netscape Messaging Server Log Sample

```
[08/Jan/2002:11:30:00 +0100] rodolf smtpd[29296]: \
    General Information: Log created (1010485800)
[08/Jan/2002:11:30:00 +0100] rodolf smtpd[29296]: \
    General Notice: SMTP-Accept:GPM7U000.J7C:\
    <john.doe.1@1.mail.example.com>:[10.0.0.1]:1.example.com.fr:\
    <john.doe.2@1.mail.example.com>:4111:1:<john.doe.3@2.mail.example.com>
[08/Jan/2002:11:30:39 +0100] rodolf smtpd[29296]: \
    General Notice: SMTP-Accept:GPM7V300.A7C:\
    <john.doe.4@1.mail.example.com>:[10.0.0.1]:1.example.com.fr:\
    <john.doe.5@1.mail.example.com>:59347:1:<john.doe.6@2.mail.example.com>
[08/Jan/2002:11:31:09 +0100] rodolf smtpd[29296]: \
    General Notice: SMTP-Accept:GPM7VX00.67E:\
    <john.doe.7@3.mail.example.com>:[10.0.0.1]:1.example.com.fr:\
    <john.doe.8@4.mail.example.com>:4117:1:<john.doe.9@2.mail.example.com>
[08/Jan/2002:11:31:26 +0100] rodolf smtpd[29296]: \
    General Notice: SMTP-Accept:GPM7WE00.D7U:\
    <john.doe.10@5.mail.example.com> (added by 2.example.com.fr):\
    [10.0.0.1]:1.example.com.fr:<john.doe.11@6.mail.example.com>:3278:1:\
    <john.doe.12@2.mail.example.com>
[08/Jan/2002:11:31:33 +0100] rodolf smtpd[29296]: \
    General Notice: SMTP-Accept:GPM7WL00.F86:
    <john.doe.13@7.mail.example.com>:[10.0.0.1]:1.example.com.fr:\
    <john.doe.14@1.mail.example.com>:998:1:<john.doe.15@2.mail.example.com>
```

Postfix

Postfix logs its information with **syslog**. No special configuration is necessary.

Example 9-4. Postfix Log Sample

```
Dec 1 04:02:56 internetsrv postfix/pickup[20919]: 693A3578E: uid=0 from=<root>
Dec 1 04:02:56 internetsrv postfix/cleanup[20921]: 693A3578E: \
    message-id=<john.doe.1@example.com>
Dec 1 04:02:57 internetsrv postfix/qmgr[20164]: 693A3578E: \
    from=<john.doe.2@example.com>, size=617 (queue active)
Dec 1 04:02:57 internetsrv postfix/cleanup[20921]: E325C578D: \
    message-id=<john.doe.1@example.com>
Dec 1 04:02:58 internetsrv postfix/local[20924]: 693A3578E: \
    to=<john.doe.2@example.com>, relay=local, delay=3, \
    status=sent (forwarded as E325C578D)
Dec 1 04:02:58 internetsrv postfix/qmgr[20164]: E325C578D: \
    from=<john.doe.2@example.com>, size=769 (queue active)
```

Qmail

Lire accepts qmail-send Qmail log files where each line starts with the timestamp in numerical (with fraction) format: `982584201.511524`. qmail-smtpd logfiles are not (yet) supported.

Tip: If you use **multilog**, you will have to filter your log file through **tai64nfraq**.

Tip: If you redirect your Qmail logs to **syslog**, you can run **lr_desyslog** (included in Lire) to remove the extra **syslog** timestamp:

```
$  lr_desyslog qmail < qmail-syslog.log > qmail.log
```

Example 9-5. Qmail Log Sample

```
998545829.342079 new msg 6416
998545829.342350 info msg 6416: bytes 2657 from \
    <bounce-debian-hurd=john.doe-debian-hurd=john.doe.1@1.mail.example.com> \
    qp 22423 uid 71
998545829.356889 starting delivery 1808: msg 6416 to local \
    john.doe.2@2.mail.example.com
```

```
998545829.357096 status: local 1/10 remote 0/20
998545829.445754 delivery 1808: success: did_0+0+1/
998545829.445976 status: local 0/10 remote 0/20
998545829.446056 end msg 6416
998545832.186954 new msg 6416
998545832.187213 info msg 6416: bytes 1957 from \
    <dns-return-13543-john-dns=john.doe.3@3.mail.example.com> qp 22431 uid 71
998545832.196806 starting delivery 1809: msg 6416 to local \
    john.doe.4@2.mail.example.com
```

Sendmail

Sendmail logs its activity through **syslog**. You need to set your *LogLevel* to 9 or higher. Versions 8.10.x and higher of Sendmail are supported.

Example 9-6. Sendmail Log Sample

```
Oct 29 14:46:13 mailhost sendmail[19504]: alias database /etc/aliases \
    rebuilt by root
Oct 29 14:46:13 mailhost sendmail[19504]: /etc/aliases: 40 aliases, \
    longest 10 bytes, 395 bytes total
Oct 29 14:52:33 mailhost sendmail[19584]: alias database /etc/aliases \
    rebuilt by root
Oct 29 14:52:33 mailhost sendmail[19584]: /etc/aliases: 40 aliases, \
    longest 10 bytes, 395 bytes total
Oct 29 15:00:00 mailhost sendmail[19633]: f9U000Y19633: from=root, \
    size=257, class=0, nrcpts=1, msgid=<john.doe.1@1.mail.example.com>, \
    relay=john.doe.2@2.mail.example.com
Oct 29 15:00:00 mailhost sendmail[19633]: f9U000Y19633: to=root, \
    ctladdr=root (0/0), delay=00:00:00, xdelay=00:00:00, mailer=local, \
    pri=30257, dsn=2.0.0, stat=Sent
Oct 29 16:00:00 mailhost sendmail[19672]: f9U100619672: from=root, size=257, \
    class=0, nrcpts=1, msgid=<john.doe.3@1.mail.example.com>, \
    relay=john.doe.2@2.mail.example.com
Oct 29 16:00:00 mailhost sendmail[19672]: f9U100619672: to=root, \
    ctladdr=root (0/0), delay=00:00:00, xdelay=00:00:00, mailer=local, \
    pri=30257, dsn=2.0.0, stat=Sent
Oct 29 17:00:00 mailhost sendmail[19696]: f9U200V19696: from=root, \
    size=257, class=0, nrcpts=1, msgid=<john.doe.4@1.mail.example.com>, \
    relay=john.doe.2@2.mail.example.com
Oct 29 17:00:00 mailhost sendmail[19696]: f9U200V19696: to=root, \
    ctladdr=root (0/0), delay=00:00:00, xdelay=00:00:00, mailer=local, \
    pri=30257, dsn=2.0.0, stat=Sent
```

Chapter 10. Firewall Supported Log Formats

Lire supports logs from many packet filter firewalls.

Cisco ACL

Cisco routers that use IOS can log activity via **syslog**. Lire is able to process the logs entries corresponding to the packet filters.

Example 10-1. IOS Log Sample

```
Aug 19 04:02:34 1.example.com.nl 218963: Aug 19 04:02:32.977: \
    %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed \
    state to down
Aug 19 04:02:34 1.example.com.nl 218964: Aug 19 04:02:33.262: \
    %ISDN-6-DISCONNECT: Interface BRI0:1  disconnected from \
    172605440 teraar, call lasted 42 seconds
Aug 19 04:02:35 1.example.com.nl 218965: Aug 19 04:02:33.266: \
    %LINK-3-UPDOWN: Interface BRI0:1, changed state to down
Aug 19 04:02:38 1.example.com.nl 218966: Aug 19 04:02:36.103: \
    %SEC-6-IPACCESSLOGP: list 102 denied tcp 10.0.0.1(4652) -> \
    10.0.0.2(80), 1 packet
Aug 19 04:02:45 1.example.com.nl 218967: Aug 19 04:02:43.543: \
    %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0, TEI 86 changed to down
Aug 19 04:02:53 1.example.com.nl 218968: Aug 19 04:02:51.471: \
    %SEC-6-IPACCESSLOGP: list 102 denied tcp 10.0.0.3(2162) -> \
    10.0.0.4(80), 1 packet
Aug 19 04:03:06 1.example.com.nl 218969: Aug 19 04:03:04.585: \
    %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0, TEI 86 changed to down
Aug 19 04:03:10 1.example.com.nl 218970: Aug 19 04:03:08.867: \
    %SEC-6-IPACCESSLOGP: list 102 denied tcp 10.0.0.5(2342) -> \
    10.0.0.6(80), 1 packet
Aug 19 04:03:12 1.example.com.nl 218971: Aug 19 04:03:10.771: \
    %SEC-6-IPACCESSLOGP: list 102 denied tcp 10.0.0.7(1093) -> \
    10.0.0.8(80), 1 packet
Aug 19 04:03:36 1.example.com.nl 218972: Aug 19 04:03:34.373: \
    %SEC-6-IPACCESSLOGP: list 102 denied tcp 10.0.0.9(3173) -> \
    10.0.0.10(80), 1 packet
```

IPChains

IPChains will log packets marked for logging through\ **syslog** (actually the kernel log buffer which is usually sent to **syslog**). Lire expects the logs in the form of a syslog log file.

Example 10-2. IPChains Log Sample

```
Oct 28 04:02:30 firewall kernel: Packet log: output DENY eth0 PROTO=17 \
    10.0.0.1:137 10.0.0.2:137 L=78 S=0x00 I=36930 F=0x0000 T=64 (#7)
Oct 28 04:07:30 firewall kernel: Packet log: output DENY eth0 PROTO=17 \
    10.0.0.1:137 10.0.0.2:137 L=78 S=0x00 I=37211 F=0x0000 T=64 (#7)
Oct 28 04:07:40 firewall kernel: Packet log: input DENY eth1 PROTO=17 \
    10.0.0.3:138 10.0.0.4:138 L=256 S=0x00 I=37213 F=0x0000 T=64 (#7)
Oct 28 04:07:40 firewall kernel: Packet log: input DENY eth1 PROTO=17 \
    10.0.0.3:138 10.0.0.4:138 L=236 S=0x00 I=37214 F=0x0000 T=64 (#7)
Oct 28 04:08:20 firewall kernel: Packet log: output DENY lo PROTO=17 \
    10.0.0.5:138 10.0.0.2:138 L=256 S=0x00 I=37216 F=0x0000 T=64 (#7)
Oct 28 04:12:30 firewall kernel: Packet log: output DENY eth0 PROTO=17 \
    10.0.0.1:137 10.0.0.2:137 L=78 S=0x00 I=37255 F=0x0000 T=64 (#7)
Oct 28 04:17:30 firewall kernel: Packet log: output DENY eth0 PROTO=17 \
    10.0.0.1:137 10.0.0.2:137 L=78 S=0x00 I=37364 F=0x0000 T=64 (#7)
Oct 28 04:19:40 firewall kernel: Packet log: input DENY eth1 PROTO=17 \
    10.0.0.3:138 10.0.0.4:138 L=256 S=0x00 I=37440 F=0x0000 T=64 (#7)
Oct 28 04:19:40 firewall kernel: Packet log: input DENY eth1 PROTO=17 \
    10.0.0.3:138 10.0.0.4:138 L=236 S=0x00 I=37441 F=0x0000 T=64 (#7)
Oct 28 04:20:20 firewall kernel: Packet log: output DENY lo PROTO=17 \
    10.0.0.5:138 10.0.0.2:138 L=256 S=0x00 I=37453 F=0x0000 T=64 (#7)
```

IP Filter

IP Filter logs selected packets through **syslog**.

Example 10-3. IP Filter Log Sample

```
Oct 30 07:42:29 firewall ipmon[16747]: 07:42:28.585962              ie0 @0:9 \
  b 192.168.48.1,45085 -> 192.168.48.2,22 PR tcp len 20 64 -S OUT
Oct 30 07:40:24 firewall ipmon[16747]: 07:40:23.631307              ep1 @0:6 \
  b 192.168.26.5,113 -> 192.168.26.1,3717 PR tcp len 20 40 -AR OUT
Oct 30 07:42:29 firewall ipmon[16747]: 07:42:28.585962              ie0 @0:9 \
  b 192.168.48.1,45085 -> 192.168.48.2,22 PR tcp len 20 64 -S OUT
Oct 30 07:44:11 firewall ipmon[16747]: 07:44:10.605416 2x              ep1 @0:15 \
  b 192.168.26.1,138 -> 192.168.26.255,138 PR udp len 20 257  IN
Oct 30 07:44:34 firewall ipmon[16747]: 07:44:33.891869              ie0 @0:10 \
  b 192.168.48.1,23406 -> 192.168.48.2,22 PR tcp len 20 64 -S OUT
```

IPTables

IPTables will log packets marked for logging through **syslog** (actually the kernel log buffer which is usually sent to **syslog**). Lire expects the logs in the form of a syslog log file.

A problem with logs from IPTables is that we have no real idea of what happened with the packet (was it denied or permitted). The logging module of IPtables permit to tag each logged packet with a prefix. Lire will interpret packets having a prefix which contains the strings `denied`, `drop`, `deny` or `reject` as denied packets. All other packets will have an unknown action value (–).

Example 10-4. IPTables Log Sample

```
Sep 21 11:45:17 lire kernel:  Packet-drop  IN=eth0 OUT=eth0 SRC=10.0.0.1 \
    DST=10.0.0.2 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=38365 DF \
    PROTO=TCP SPT=3117 DPT=80 WINDOW=16384 RES=0x00 SYN URGP=0
Sep 21 11:45:20 lire kernel:  Packet-drop  IN=eth0 OUT=eth0 SRC=10.0.0.1 \
    DST=10.0.0.2 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=38478 DF \
    PROTO=TCP SPT=3117 DPT=80 WINDOW=16384 RES=0x00 SYN URGP=0
Sep 21 11:45:26 lire kernel:  Packet-drop  IN=eth0 OUT=eth0 SRC=10.0.0.1 \
    DST=10.0.0.2 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=38680 DF \
    PROTO=TCP SPT=3117 DPT=80 WINDOW=16384 RES=0x00 SYN URGP=0
Sep 21 11:52:46 lire kernel:  Packet-drop  IN=eth0 OUT=eth0 SRC=10.0.0.1 \
    DST=10.0.0.3 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=54122 DF \
    PROTO=TCP SPT=4532 DPT=80 WINDOW=16384 RES=0x00 SYN URGP=0
Sep 21 11:52:49 lire kernel:  Packet-drop  IN=eth0 OUT=eth0 SRC=10.0.0.1 \
    DST=10.0.0.3 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=54222 DF \
    PROTO=TCP SPT=4532 DPT=80 WINDOW=16384 RES=0x00 SYN URGP=0
Sep 21 11:52:55 lire kernel:  Packet-drop  IN=eth0 OUT=eth0 SRC=10.0.0.1 \
    DST=10.0.0.3 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=54443 DF \
    PROTO=TCP SPT=4532 DPT=80 WINDOW=16384 RES=0x00 SYN URGP=0
```

WebTrends Enhanced Log Format

The WELF format is a format developed by WebTrends and supported by many firewall vendors. Products can save log files in that format directly or can log through **syslog**. Either native WELF log files or **syslog**'s log files contain WELF information. Although the log format isn't designed for packet filter firewalls (it can contain information from devices that do network intrusion or proxy services), Lire does its best to map this information to something that can be meaningful.

Example 10-5. WELF Log Sample

```
WTsyslog[1998-08-01 14:05:46 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 04:10:23" fw=WebTrendsSample pri=5 \
    msg="ICMP packet dropped" src=10.0.0.2 dst=10.0.0.3 rule=3
WTsyslog[1998-08-01 16:31:00 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 10:35:38" fw=WebTrendsSample pri=6 \
    proto=tcp/443 src=10.0.0.4 dst=10.0.0.5 rcvd=4844
WTsyslog[1998-08-01 16:31:01 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 10:35:38" fw=WebTrendsSample pri=6 proto=tcp/443 \
    src=10.0.0.4 dst=10.0.0.5 rcvd=6601
```

```
WTsyslog[1998-08-01 16:43:59 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 10:48:36" fw=WebTrendsSample pri=5 \
    msg="UDP packet dropped" src=10.0.0.6 dst=10.0.0.3 rule=3
WTsyslog[1998-08-01 16:46:13 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 10:50:50" fw=WebTrendsSample pri=5 \
    msg="UDP packet dropped" src=10.0.0.7 dst=10.0.0.3 rule=3
WTsyslog[1998-08-01 16:46:13 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 10:50:50" fw=WebTrendsSample pri=6 proto=telnet \
    src=10.0.0.4 dst=10.0.0.8 sent=1194
```

Lire also supports some extension uses by SonicWall.

Example 10-6. SonicWall Log Sample

```
Jan  7 15:01:10 lire id=firewall sn=asdlFFFXSD \
    time="2002-01-06 22:42:13" fw=10.0.0.1 pri=6 c=1 m=30 \
    msg="Administrator login failed - incorrect password" n=1 \
    src=10.0.0.2:LAN dst=10.0.0.1
Jan  7 15:01:16 lire id=firewall sn=asdlFFFXSD \
    time="2002-01-06 22:42:19" fw=10.0.0.1 pri=6 c=1 m=29 \
    msg="Successful administrator login" n=1 src=10.0.0.2:LAN dst=10.0.0.1
Jan  7 15:02:32 lire id=firewall sn=asdlFFFXSD \
    time="2002-01-06 22:43:34" fw=10.0.0.1 pri=5 c=128 m=37 \
    msg="UDP packet dropped" n=1 src=10.0.0.3:68 dst=10.0.0.4:67 dstname=DHCP
Jan  7 15:31:43 lire id=firewall time="2002-01-07 15:20:21" \
    fw=10.0.0.5 pri=6 proto=dns src=10.0.0.6 dst=10.0.0.8 rcvd=130 \
    sn=asdlFFFXSD 54 c=1024 m=98 n=31
Jan  7 15:31:43 10.0.0.5 id=firewall time="2002-01-07 15:20:21" \
    fw=10.0.0.5 pri=6 proto=dns src=10.0.0.6 dst=10.0.0.9 rcvd=130 \
    sn=asdlFFFXSD 54 c=1024 m=98 n=32
```

Chapter 11. FTP Supported Log Formats

Lire supports the widely used `xferlog` FTP file transfer log files and logs from the FTP service of Microsoft Internet Information Server.

Microsoft Internet Information Server

The FTP log file from Microsoft Internet Information Server is a variant of the W3C Extended Log Format defined at http://www.w3.org/TR/WD-logfile.html.

Lire can use the following fields of the format: *`date`*, *`time`*, *`c-ip`*, *`c-dns`*, *`cs-bytes`*, *`time-taken`*, *`cs-uri-stem`* and *`cs-method`*. The other fields will be ignored.

Example 11-1. Microsoft Internet Information Server FTP Log Sample

```
#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2001-11-29 00:01:32
#Fields: time c-ip cs-method cs-uri-stem sc-status
00:01:32 10.0.0.1 [56]created spacedat/091001092951LGW_Data.zip 226
00:01:32 10.0.0.1 [56]created spacedat/html/bx01g01.gif 226
00:01:32 10.0.0.1 [56]created spacedat/html/catlogo.gif 226
00:01:32 10.0.0.1 [56]QUIT - 226
00:03:32 10.0.0.1 [58]USER badm 331
00:03:32 10.0.0.1 [58]PASS - 230
```

Xferlog

The `xferlog` format is supported by a wide range of FTP servers like Wu-Ftpd, ProFTPD or standard BSD **ftpd**.

Example 11-2. Xferlog Log Sample

```
Mon Feb 26 09:48:18 2001 1 1.example.com 147456 \
    /var/ftp/pubinfo/sm2/esc/s82e5937.jpg b _ o a \
    john.doe.1@1.mail.example.com ftp 0 * i
Mon Feb 26 10:26:31 2001 1 2.example.com 10593 \
    /var/html/public/htdocs/pubinfo/pr/1999/28/extra-photos.html \
    a _ i r kellys ftp 0 * c
Mon Feb 26 10:27:50 2001 1 2.example.com 14 \
    /var/html/public/htdocs/pubinfo/pr/1999/28/extra-photos.html.LCK \
    a _ i r kellys ftp 0 * c
Mon Feb 26 10:28:17 2001 1 2.example.com 14 \
    /var/html/public/htdocs/pubinfo/pr/1999/28/extra-photos.html.LCK \
```

```
    a _ o r kellys ftp 0 * c
Mon Feb 26 10:28:18 2001 1 2.example.com 10591 \
    /var/html/public/htdocs/pubinfo/pr/1999/28/extra-photos.html \
    a _ i r kellys ftp 0 * c
Mon Feb 26 12:51:02 2001 2 3.example.com 43063 \
    /var/ftp/pubinfo/jpeg/EtaCar3d.jpg  b _ o a mozilla@ ftp 0 * c
Mon Feb 26 12:51:17 2001 2 3.example.com 37332 \
    /var/ftp/pubinfo/jpeg/EtaCarC.jpg b _ o a mozilla@ ftp 0 * c
Mon Feb 26 12:51:52 2001 6 3.example.com 62823 \
    /var/ftp/pubinfo/jpeg/EtaCarD.jpg b _ o a mozilla@ ftp 0 * c
Mon Feb 26 12:52:31 2001 2 3.example.com 33660 \
    /var/ftp/pubinfo/jpeg/Neptune.jpg b _ o a mozilla@ ftp 0 * c
Mon Feb 26 12:52:43 2001 2 3.example.com 26295 \
    /var/ftp/pubinfo/jpeg/NeptDS.jpg b _ o a mozilla@ ftp 0 * c
```

Chapter 12. Message Store Supported Log Formats

Lire supports log files from Netscape Messaging Server and Netsape Messaging Server Mail Multi Plexor.

Chapter 13. Print Supported Log Formats

The print superservice supports printer logs from two print daemons.

CUPS page_log

Information about this format can be found in the CUPS Software Administrators Manual (http://www.cups.org/sam.html).

Example 13-1. CUPS page_log Log Sample

```
DANKA_infotec_P450 kurt 137 [19/Aug/2001:16:58:58 +0100] 1 1
P450l kurt 138 [19/Aug/2001:17:05:06 +0100] 1 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 2 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 3 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 4 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 5 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 6 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 7 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 8 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 9 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 10 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 11 1
P450l kurt 138 [19/Aug/2001:17:05:08 +0100] 12 1
```

LPRng Account Log File

Example 13-2. LPRng Log Sample

Lire can process the accounting file associated with a LPRng print queue. The format of the file is described at http://www.lprng.org/LPRng-HOWTO-Multipart/x9481.htm

```
jobstart '-Hh4.private' '-nroot' '-Pps' '-kcfA938h4.private' \
    '-b1093' '-tNov  5 19:39:25'
start '-p12942' '-kcfA938h4.private' '-nroot' '-hh4.private' '-Pps' \
    '-c0' '-Fo' '-tSun Nov  5 19:39:25 1995'
filestart '-p12944' '-kcfA938h4.private' '-nroot' '-hh4.private' '-Pps' \
    '-c0' '-Ff' '-tSun Nov  5 19:39:27 1995'
fileend '-p12944' '-kcfA938h4.private' '-nroot' '-hh4.private' '-Pps' \
    '-b3' '-c0' '-Ff' '-tSun Nov  5 19:39:58 1995'
end '-p12942' '-kcfA938h4.private' '-nroot' '-hh4.private' '-Pps' \
    '-b2' '-c0' '-Fo' '-tSun Nov  5 19:39:59 1995'
jobend '-Hh4.private' '-nroot' '-Pps' '-kcfA938h4.private' \
    '-b1093' '-tNov  5 19:39:59'
```

Chapter 14. Proxy Supported Log Formats

Lire supports three different proxy log file formats allowing it to support a wide range of products.

Microsoft Internet Security and Acceleration Server

This product uses a format derived from the W3C Extended Log Format which is defined at
http://www.w3.org/TR/WD-logfile.html. Information about the way
Microsoft Internet Security and Acceleration Server uses that format can be found on the product's website
(http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/isa/proddocs/isadocs/M_S_C_LoggingFi

The format of

Lire can use the following fields of the format: *date*, *time*, *c-ip*, *c-host*, *cs-username*, *c-agent*, *time-taken*, *r-ip*, *r-host*, *sc-status*, *sc-protocol*, *sc-operation*, *s-object-source*, *sc-operation*, *rule#1*, *rule#2* and *cs-mime-type*. The other fields will be ignored.

Example 14-1. Microsoft Internet Security and Acceleration Server Log Sample

```
#Software: Microsoft(R) Internet Security and Acceleration Server 2000
#Version: 1.0
#Date: 2002-01-16 07:00:01
#Fields: c-ip cs-username c-agent date time s-computername \
    cs-referred r-host r-ip r-port time-taken cs-bytes\
    sc-bytes cs-protocol s-operation cs-uri s-object-source \
    sc-status
10.0.0.1 anonymous Mozilla/4.0 (compatible; MSIE 5.0; Win32)\
    2002-01-16 07:00:01 GRO1SYX01 - - - -\
    - 155 2569 - GET - - 200 \
10.0.0.1 anonymous Outlook Express/5.0 \
    (MSIE 5.0; Windows 98; DigExt) 2002-01-16 07:00:04 \
    GRO1SYX01 - 1.example.com
```

Squid

Lire can process native Squid access logs.

Example 14-2. Squid Log Sample

```
1011164724.171   1337 10.0.0.1 TCP_MISS/200 20110 GET \
    http://images.google.com/images? - DIRECT/10.0.0.2 text/html
1011164724.965    740 10.0.0.1 TCP_MISS/200 26461 GET \
    http://www.ia.hiof.no/informatikk/forelesning/historie/historie.html \
    - DIRECT/10.0.0.3 text/html
```

```
1011164727.626   2580 10.0.0.1 TCP_MISS/200 111927 GET \
    http://www.ia.hiof.no/informatikk/forelesning/historie/transistor.jpg \
    - DIRECT/10.0.0.3 image/jpeg
1011164731.619    687 10.0.0.1 TCP_MISS/200 18191 GET \
    http://images.google.com/images? - DIRECT/10.0.0.2 text/html
1011164734.972   3282 10.0.0.1 TCP_MISS/200 29595 GET \
    http://www.hillnews.com/restaurants/rst_tosca.shtm - \
    DIRECT/10.0.0.4 text/html
1011164735.482    467 10.0.0.1 TCP_MISS/200 7839 GET \
    http://www.hillnews.com/global/banner_logo.gif - \
    DIRECT/10.0.0.4 image/gif
1011164740.163   1004 10.0.0.1 TCP_MISS/200 19580 GET \
    http://images.google.com/images? - DIRECT/10.0.0.2 text/html
1011164741.905   1687 10.0.0.1 TCP_MISS/200 17383 GET \
    http://www.charlotteregional.com/speech.html - DIRECT/10.0.0.5 text/html
1011164742.214    275 10.0.0.1 TCP_MISS/200 8001 GET \
    http://www.charlotteregional.com/images/st2.jpg - \
    DIRECT/10.0.0.5 image/jpeg
1011164745.891    716 10.0.0.1 TCP_MISS/200 18796 GET \
    http://images.google.com/images? - DIRECT/10.0.0.2 text/html
```

WebTrends Enhanced Format

The WELF format is a format developed by WebTrends and supported by many firewall vendors. Products can save log files in that format directly or can log through **syslog**. Either the WELF log files or **syslog**'s log files contain WELF information. This format can be used by packet filter firewalls, proxies or network intrusion detection devices. This Lire superservice will only process records that are related to proxy services (either application proxy like a web proxy or a transport proxy like for the telnet protocol).

Example 14-3. WELF Log Sample

```
WTsyslog[1998-08-01 00:04:11 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 00:08:52" fw=WebTrendsSample pri=6 proto=http \
    src=10.0.0.2 dst=10.0.0.3 dstname=1.example.com \
    arg=/selfupd/x86/en/WULPROTO.CAB op=GET result=304 sent=898
WTsyslog[1998-08-01 00:04:12 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 00:08:52" fw=WebTrendsSample pri=6 proto=http \
    src=10.0.0.2 dst=10.0.0.3 dstname=1.example.com \
    arg=/selfupd/x86/en/CUNPROT2.CAB op=GET result=304 sent=853
WTsyslog[1998-08-01 00:04:23 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 00:09:03" fw=WebTrendsSample pri=6 proto=http \
    src=10.0.0.2 dst=10.0.0.3 dstname=1.example.com \
    arg=/R510/v31content/90820/0x00000409.gng op=GET result=304 sent=2983
WTsyslog[1998-08-01 03:02:03 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 03:06:43" fw=WebTrendsSample pri=6 proto=http \
    src=10.0.0.2 dst=10.0.0.4 dstname=2.example.com arg=/ op=POST \
```

```
    result=200 sent=2195
WTsyslog[1998-08-01 16:25:33 ip=10.0.0.1 pri=6] id=firewall \
    time="1998-08-01 06:30:09" fw=WebTrendsSample pri=6 proto=http \
    src=10.0.0.5 dst=10.0.0.6 dstname=3.example.com \
    arg=/portal/brand/images/logo_pimg.gif op=GET result=304 rcvd=1036
```

Chapter 15. Syslog Supported Log Formats

Lire supports more than 7 different syslog file formats.

Chapter 16. WWW Supported Log Format

The WWW superservice supports four log file formats which makes it possible to support a wide range of web servers like Apache, IIS or Boa.

Common Log Format

Common Log Format (CLF) is a standard log format that was originally implemented in the CERN httpd web server but that is supported nowadays by most web servers. Apache, IIS and Boa can be configured to log in that format.

The Common Log Format has the following format:

remotehost rfc931 authuser [*date*] "*request*" *status bytes*

where the fields have the following meaning:

remotehost

The host that made the request. This can be given as an IP address or a hostname.

rfc931

The result of an ident lookup on the host. This is usually not used.

authuser

The authenticated username.

date

The timestamp of the request.

request

The first line of the HTTP request. Usually in the format "*method file protocol*".

status

The result status of the request. i.e. `200`, `301`, `404`, `500`.

bytes

The size of the response sent back to the client.

Example of log lines in Common Log Format :

```
127.0.01 - - [11/03/2001 12:12:01 -0400] "GET / HTTP/1.0" 200 513
dsl1.myprovider.com - francis [11/03/2001 12:14:01 -0400] \
"GET /secret/ HTTP/1.0" 200 1256
```

Combined Log Format

The combined log format is an extension to the Common Log Format. It adds informations about the user agent and referer. It is also known as the extended common log format. It was first implemented in the NSCA httpd web server but is now supported in many web servers. Apache can be configured to use this log format.

Two fields are added at the end of the common log lines:

"*referer*" "*useragent*"

referer

The content of the Referer header of the request. This usually reflects the page the user visited before this request.

useragent

The content of the User-Agent header of the request. This usually reflects the browser that the user is using.

CLF With mod_gzip Extensions

Mod_gzip is another extension to the common log format. It is used by the mod_gzip Apache extension which can be used to compress the result of requests before sending them to the client.

mod_gzip is a module developed by RemoteCommunications, Inc. Sourcecode is freely available from http://www.RemoteCommunications.com/apache/mod_gzip/mod_gzip. More informations can be found in their FAQ (http://www.RemoteCommunications.com/apache/mod_gzip/mod_gzip_faq.htm).

mod_gzip can log information about the compression of pages. To enable this, one can configure Apache to log using the 'gzip' format which can be defined as follows:

```
LogFormat "%h %l %u %t \"%r\" %>s %b %{mod_gzip_result}n \
        %{mod_gzip_compression_ratio}n" gzip
```

This adds two fields at the end of each common log line:

gzip_result compression_ratio

gzip_result

The **gzip** result code. Usually `OK`.

compression_ratio

The ratio by which the content was compressed. A number from 0 to 100.

Referer Log Format

The Referer log format is an old format that was implemented in the NSCA httpd server. It was used to log information about the request's referer in a separate log file. The combined log format has made this log format obsolete.

Referer log files have the following format:

uridocument

uri

The referring URI. This is the content of the Referer header of the request which usually reflects the page where the user was before that request.

document

The local document that was referenced by that URI. This is the requested file without any query string.

Logs With Virtual Host Information

You may encounter log files that have a field containing the virtual host for which the requests was at the beginning of the line. The rest of the line is usually in the common or combined log format. This kind of logging is typically seen on webservers hosting several virtual servers.

Example of such a line:

```
www.example.com 1.7.2.21 - - [13/Oct/2000:10:30:16 +0200] \
  "GET / HTTP/1.0" 200 83
```

Although Lire doesn't directly support such logs, it is easy to split those logs into many log files in the common or combined log format which can subsequently be processed by Lire.

Example doing this in a shell:

```
$  mkdir apache-common.log
$  (while read virt rest; do echo $rest >> \
apache-common.log/$virt; done) < /var/log/apache/common.log
$  for f in apache-common.log/*; do \
lr_log2mail -s "$f" common joe@example.com < $f; done
```

W3C Extended Log Format

This is a log format defined by the W3C which can contain a variable amount of information. The format is defined at http://www.w3.org/TR/WD-logfile.html.

This log format uses a header to specify the order of the fields present in the log file.

Lire can use the following fields of the format: `date`, `time`, `c-ip`, `c-dns`, `cs-uri`, `cs-method`, `sc-bytes`, `sc-status`, `cs(User-Agent)`, `cs(Referer)`, `cs-uri-stem` and `cs-username`. The other fields will be ignored.

III. Lire Reference

Chapter 17. Installation Parameters

This chapter describes the various configuration variables that can be set when installing Lire. These can be set using options to **./configure** or by setting environment variables.

./configure parameters

`--prefix`

This option specifies where Lire will be installed.

Defaults to `/usr/local`.

`--bindir`

This option specifies where Lire's executables intended for users will be installed.

Defaults to *`${prefix}`*`/bin`.

`--sysconfdir`

This option specifies where Lire's configuration files will be installed. (Actually, they will be installed in a subdirectory named `lire`.)

Defaults to *`${prefix}`*`/etc`.

`--libexecdir`

This option specifies where Lire's internal executables and scripts will be installed. (Actually, they will be installed in a subdirectory of this one named `lire`.)

Defaults to *`${prefix}`*`/libexec`.

`--sharedstatedir`

This option specifies where Lire's data files will be installed. (Actually, they will be installed in a subdirectory of this one named `lire`.)

Defaults to *`${prefix}`*`/share`.

`--mandir`

This option specifies where Lire's man pages will be installed.

Defaults to *`${prefix}`*`/man`.

`--with-perl5libdir`

This option specifies where Lire's perl modules will be installed.

Defaults to *`${prefix}`*`/share/perl5`.

`--with-spooldir`

This option specifies the default value of `lr_spool_dir` which is the spool directory used by the responder. Unless you're running your own responder this variable is not relevant.

`--with-archivedir`

When you're archiving your reports and logs using the archive feature this sets the default value of `lr_archive_dir`.

Installation Environment Variables

Some environment variables can be set before running **./configure** to tune the installation process. This can be used to specify the locations of components which are installed but can't be found by **./configure** in "standard" locations. For example, you could pass the location of the DocBook DTD by running **./configure** as:

```
$  DBK_XML_DTD=/home/flacoste/xml/docbook-xml-4.1.2/docbookx.dtd \
 ./configure
```

The following list explains the purpose of each variable.

`PERL`

Sets the path to the **perl** interpreter.

`JADE`

Sets the path to the **jade** DSSSL interpreter. This is only needed when you build from CVS.

`PDFJADETEX`

Sets the path to the **pdfjatetex** command. This is only needed when you build from CVS.

`XSLTPROC`

Sets the path to the **xsltproc** XSLT processor. This is only needed when you build from CVS.

`DBK_XML_DTD`

Sets the path to the DocBook XML Document Type Declaration. This should point to the XML V4.1.2 DTD. This is only needed when you build from CVS.

`DBK_XSL_STYLESHEETS`

Sets the path to the directory which contains Norman Walsh's XSL stylesheets for DocBook. (This directory should contain subdirectories named `fo`, `html` or `xhtml`.) This is only needed when you build from CVS.

`DBK_DSSSL_STYLESHEETS`

Sets the path to the directory which contains Norman Walsh's DSSSL stylesheets for DocBook. (This directory should contain a subdirectory named `print`.) This is only needed when you build from CVS.

Chapter 18. Lire Logging and Error Messages

Logging

The Lire responder can log its messages, and output them to either standard error (stderr) or to syslog using the **logger** program. Choosing between either one of them is done with the `lr_logging_method` configuration variable.

Log Messages

Each log message has a level, which is one of:

emerg
: system is unusable

alert
: action must be taken immediately

crit
: critical conditions

err
: error conditions

warning
: warning conditions

notice
: normal, but significant, condition

info
: informational message

debug
: debug-level message

See also syslog(3).

A complete Lire message looks like

```
superservice service lr_tag program level message
```

where program is the name of the script producing the message. lr_tag is used to track different Lire jobs. E.g.

```
www apache lr_tag-20010826081801-31102 lr_log2mail notice storing \
 /tmp/lr_log2mail.apache.lr_tag-20010826081801-31102.report in \
 /var/lib/lire/data/report/ascii/www/apache/complete/example.com_20010826/2001081
```