Lire User's Manual

Joost van Baal

Egon L. Willighagen

Francis J. Lacoste

Lire User's Manual

by Joost van Baal, Egon L. Willighagen, and Francis J. Lacoste

Revision History

Revision 20020214 $Date: 2002/02/13 21:57:27 $
$Id: user-manual.dbx,v 1.28 2002/02/13 21:57:27 flacoste Exp $

Table of Contents

List of Tables

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 that 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 in 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 contains comprehensive informations on all the reports that can be generated by Lire. It describes all the supported log files and gives the descriptions and customizable parameters for each report.

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

Conventions Used

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 details 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 by sending an empty email with a subject of *subscribe* to <`questions-request@logreport.org`>. Email for the list should be sent to <`questions@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@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@logreport.org`>. Email to the list should be sent to <`development@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 or firewalls, and supports multiple output format. Lire is developed by the Stichting LogReport Foundation, more informations about the project can be found on http://www.logreport.org/.

Lire is built around the concept of *superservice*. A superservice is a class of applications which share the same reports. Lire supports 6 superservices: 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 can run in an online responder setup, as a client, as a cron driven system, or as a command line driven system. In an online responder setup, the Lire system receives emails containing log files from other hosts, and sends generated reports back by email. In a client setup, the system sends log files by email to another Lire system which runs an online responder, and receives reports back. Optionally, the log files can be anonimized before being sent. A cron driven setup reads and processes log files after they're rotated, on the local host. In a command line driven system, users run the Lire scripts on an ad-hoc basis.

Supported Systems

The package is reported to be useable on

- GNU/Linux (Debian GNU/Linux ("potato" and "woody"), Red Hat Linux (7.0, 7.1, 7.2), Mandrake Linux (7.0 Air, 7.2 Odyssey))
- BSD (FreeBSD (4.1-STABLE, 4.5-PRERELEASE, 4.4-STABLE), OpenBSD (2.7, 2.8 and 2.9), Mac OS X v 10.1)
- Solaris (SunOS 5.6 and 5.7)
- HP-UX (11.11)
- And yes, it even runs on GNU/Hurd!

Supported Applications

Lire can generate reports for a variety of dns, email, ftp and web servers as well as some firewalls. Here are the applications (services) supported in each superservice.

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.

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.

Email

Four email servers are supported by Lire:

- 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/).
- Linux 2.2.X ipchains log files. http://netfilter.samba.org/ipchains/ (http://netfilter.samba.org/ipchains/).
- IPfilter log files http://coombs.anu.edu.au/~avalon/ip-filter.html (http://coombs.anu.edu.au/~avalon/ip-filter.html).
- 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 supports potentially a lot 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 behave 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 that uses a variant of the W3C Extended Log Format.

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

Print

Lire can generate reports for two print servers:

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

The reports generated will include informations 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 supports potentially a lot 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 behave with those products.

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. For sure, 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 informations like the number of requests by day, requests by browser, attack detection, top referers, etc. It is Lire's most complete report.

You will find the definite lists of reports available for each superservice in Part II: Reports Reference.

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. The following formats are supported:

ASCII

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

HTML

Lire can generate HTML reports that can be viewed in any web browser. Those reports can include charts for easy overview.

PDF

To print the reports, Lire can generate Adobe PDF output. Like the HTML reports, those can include charts for easy overview.

RTF

For Microsoft platforms, Lire can generate reports in RTF format.

> **Warning**
>
> With the wide spread availability of PDF viewers, this format is less interesting than the PDF format, but it is supported anyhow.
>
> It may be dropped in future Lire's release.

XHTML

Alternatively to the HTML output format, some may prefer reports in the new XML based XHTML. Those can be viewed in recent web browsers like Mozilla. Like most of the other reports, XHTML can include charts for easy overview.

DocBook XML

Lire can generate reports in the standard XML DTD DocBook. This intermediary format can be interesting for those who might want to customize the layouts of the reports. For example, one could generate reports in DocBook XML and use the company's stylesheets to print reports with the company's logo and standard report appearance.

LogML

LogML (http://www.cs.rpi.edu/projects/pb/WWWPal/logml.html) is a new XML based language to compress log files in smaller files. It enables data mining techniques on web log files, by building a graph representing the website. Currently, only the email and www superservices support LogML output.

Tools to view the LogML output are not publicly available yet. Watch the WWWPal download webpage (http://www.cs.rpi.edu/projects/pb/WWWPal/download.html). One can get nice results however with OpenJGraph (http://sourceforge.net/projects/openjgraph/) by using the **runsample1** program. See also the Lire examples section (http://logreport.org/lire/ex/logml.php) on the LogReport website.

> **Note:** Not all reports can be included in the LogML format. Presently, only some reports from the www and email superservices are supported. With the www superservice, the reports supported are `top-referer-page-connections`, `bytes-by-day`, `requests-by-day`, `requests-by-result`, `requests-by-method`, `requests-by-version` and `bytes-by-dir`. With the email superservice, the only report supported is `top-deliveries-btw-relays`

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 the informations regarding 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. Installation notes about specific platform (Debian GNU/Linux, Solaris, OpenBSD, etc.) can be found in the *Lire FAQ*.

Client Installation

The simplest setup to install Lire is in a client-server scenario 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 (anyone 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 5.005_03 or later. (5.00503, 5.6.0 and 5.6.1 works).
- 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/.

  As a convenience for our users, you can download a version of Lire which includes those two libraries. Many systems also have prepackaged versions, you're advised to install those packages when available. (They're both packaged for Debian GNU/Linux, as well as for OpenBSD 2.9., see the *Lire FAQ* for complete details.)

- 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 ASCII reports only.

Requirements for Other Output Formats

To generate reports in other output formats than ASCII, you will need the following additional requirements:

- An XSLT processor. Currently the only supported XSLT processor is **xsltproc**, included with the XSLT C Library for Gnome (libxslt). You need version 1.0.4 or later.

  You can download this library along with the libxml2 library which it required from http://xmlsoft.org/XSLT/.

To generate charts, you will need the following requirements:

- The GD::Graph perl module available from CPAN mirrors (http://www.cpan.org/modules/by-module/GD/).

  This module requires the GD and GD::Text perl modules as well as the libgd library.

  The GD Graphics Library is available from http://www.boutell.com/gd/.

To generate HTML or XHTML reports, you will need in addition to the XSLT processor, the following requirements:

- The DTD for DocBook XML 4.1.2. This is available from http://www.docbook.org/xml/4.1.2/index.html.
- Norman Walsh's XSL stylesheets for DocBook. You can download these stylesheets from http://docbook.sourceforge.net/projects/xsl/index.html.

To generate PDF and RTF reports, you will need the following additional requirements:

- Jade, James Clarks' engine for the DSSSL style language. You can also use OpenJade, the name under which Jade is currently being maintained and extended.

  You can download Jade from http://www.jclark.com/jade/. OpenJade is available from http:///openjade.sourceforge.net/.

- The DTD for DocBook XML 4.1.2. This is available from http://www.docbook.org/xml/4.1.2/index.html.
- Norman Walsh's DSSSL stylesheets for DocBook. You can download those stylesheets from http://docbook.sourceforge.net/projects/dsssl/index.html
- *(For PDF output only)*. JadeTeX and recent TeX installation.

  JadeTeX is available from http://jadetex.sourceforge.net/.

Other Optional Requirements

Other optional things you might 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 mirrors, http://www.cpan.org/modules/by-module/Time/. If it isn't present in the system, the required files are included with Lire and will be installed.

Installing

Installation of Lire is pretty straightforward:

1. Make sure that you have the necessary 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 should have found all the requirements you had installed.

> **Note:** For the SGML/XML components (DocBook DTD, Norman Walsh's DSSSL and XSL stylesheets), **configure** should find them if they were installed in "standard" places. (This is somewhere in a `sgml` tree as specified in the FHS and as you will find on most recent GNU/Linux distributions.) If they aren't found, you may hint **configure** by specifying their location through the use of environment variables:
>
> ```
> $ DBK_XML_DTD=path_to_docbook_dir/docbookx.dtd \
>  DBK_DSSSL_STYLESHEETS=path_to_dbk_dsssl_dir \
>  DBK_XSL_STYLESHEETS=path_to_dbk_xsl_dir \
>  ./configure [--prefix=path]
> ```
>
> Similarly, you can use other environment variables to hint for other things that Lire doesn't find. See Chapter 14 for the complete list.

4. Compile the software (if you have XML::Parser installed, this will consist only in 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 informations on using Lire. You can also read the Section called *Configuring Lire Using lr_config* to learn how to configure Lire.

Anonimized Client Installation

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

Requirements

The anonimized 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 anonimizing 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 might have to install it separately.

Installing

There is no difference between the anonimized 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*).

You will also need the following additional requirement:

- Mpack. This is used to decode MIME attachments. You can download Mpack from ftp://ftp.andrew.cmu.edu/pub/mpack/.

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 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.

One can create the maildirs by doing e.g.

```
$ cd ~/logreport
$ mkdir -p var/spool/lire/apachecommon
$ maildirmake var/spool/lire/apachecommon/Maildir
$ cd ~/logreport/var/spool/lire
$ mkdir bind8 postfix qmail sendmail
$ maildirmake bind8/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 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: , 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.) At last, do e.g.

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

One could create more maildirs, and configure the useraccount to store mails for different services in different maildirs. We wont go into this much detail here though.

Installing Under Postfix

Installing Under qmail

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

```
$ cd ~/logreport/var/spool/lire/postfix
$ maildirmake Maildir
$ echo './logreport/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 sent your qmail-send process a SIGHUP.

Making The Responder Run At Boot

Configuring Lire Using lr_config

After Lire is installed, you can configure it by running the **lr_config** command. This is an interactive program that asks questions and writes the appropriate variables based on the answers to `$sysconf/etc/lire/defaults.local`. (So you need to run the command as a user who has write access to that file). This file can later be modified by hand. (You will find in the reference part of this manual the description of all available configuration variables: see Chapter 15.)

Additionally, if you want to use the **lr_cron** command to run periodically Lire through a cron job, you will need to run **lr_config** to configure it.

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 setup your system to automatically process your log files at regular interval.

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 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 banwidth, 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 all mailers 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.
> - 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 attachment (like a signature) after the log file.

Stichting LogReport Foundation offers as a public service 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.logreport.org < \
 /var/log/query.log
```

To save bandwidth, please sent 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 anonimized log to the responder. Consult the Section called *Sending Anonimized 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. This command expects the log file on standard input and will output the report on standard output. It takes three arguments: a file where the error message will be saved, the superservice and the service of the log file. (The various values available for the superservice and service argument can be found in lr_log2report(1) man page). The **lr_log2report** command will log a lot of information on standard error. You can use the **lr_run** wrapper to filter those messages according to your preferences.

> **Note:** The file specified in the first argument will only be created if serious errors are detected. It will contain a short explanation of what went wrong and will suggest possible causes of the problem: bad service, invalid log lines, etc. This is the error message that is sent back by the responder. In interactive use, it's probably usefull only when debugging.

Example 3-2. Generating a Report With lr_log2report

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

```
$ lr_run lr_log2report /tmp/error www 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 `-o` 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_run lr_log2report -o html /tmp/error www combined < \
   /var/log/apache/access_log > ~/apache.html
```

Including Charts in the Report

Lire can generate charts (pie chart, bar chart, line graph or histogram) for some reports. If you have the necessary requirements, you can tell **lr_log2report** to make the charts by adding the `-i` option to your command line. (Only the PDF, HTML, XHTML, RTF and DocBook XML output formats support charts.)

Example 3-4. Generating A HTML Report With Charts

To include charts with the HTML report, you would use the following command:

```
$ lr_run lr_log2report -o html -i /tmp/error www combined < \
   /var/log/apache/access_log | tar xfC - /tmp
```

In this case, we piped the output to **tar** because what is outputted is a tar file. This tar file contains a directory called `report` which contains the charts in PNG format and the HTML report in the `index.html` file. The command creates a directory `/tmp/report/`.

Sending Anonimized Log Files To A Responder

For more privacy, you can anonimize your log somewhat before sending it to a responder. Lire includes a command called **lr_anonimize** which will transform everything that looks like an IP address, an email or a domain name into an anonimized 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 anonimized 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_anonimize** and make sure that the subject of your email starts with `anon`.

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

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

```
$  grep ' postfix/' /var/log/mail.log | \
    lr_run lr_anonimize /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 anonimized values.

> **Warning**
>
> **lr_anonimize** 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 deanonimize it and run the appropriate converter on the deanonimized report. The

converter for a specific output format is called **lr_xml2*`format`***. For example, you would use the **lr_xml2pdf** command to generate a PDF report.

Example 3-6. Deanonimizing 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_deanonimize /tmp/anon < /tmp/anon-report.xml > /tmp/report.xml
$  lr_run lr_xml2html /tmp/report.xml > /tmp/report.html
```

You could also generate charts by adding the `-i` to the **lr_xml2html** command.

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 box as a reporting box (or "online responder"), while other machines sent their log files to it by email for processing. (If remote syslogging is used, a cron-driven setup is sufficient.)

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

Using Syslog

Chapter 4. Automating Lire

This chapter discusses various ways to configure Lire for generating periodical reports from your system logs.

Automatically Processing Log Files Using Cron

The easiest way to have Lire generate reports from the various log files available on your system is through a **cron** job. Lire includes a script called **lr_cron** which takes care of calling the appropriate batch of commands on the appropriate log files. The reports are generated in the ASCII format and are sent to an email address of your choice. To use **lr_cron**, you are advised to configure Lire using **lr_config**.

Configuring lr_cron

When running **lr_config**, it will first ask questions about the global configuration of Lire and specific question for when you run a responder. After that, it will ask questions to configure the cronjob.

First, you will have the possibility to set default values that will be used for each service. You will also get the chance to override those defaults on a per-service case. Those parameters are: the email address that will receive the report and a string that will appear in the subject of the email.

All further questions ask you about which services you want Lire to report about. The first question for each super service is always wether you are collecting log files related to this superservice. If you answer "no", it will skip further questions about this superservice.

The configuration program will then ask you questions about which services are run one-by-one. For email, for example, these services might be exim, postfix, qmail and sendmail. For each program **lr_config** asks you where lire can find the log files. It will also ask you about a filter that should be run on the log file before processing it. This way, it can process compressed log files or can only keep the relevant lines for a particular service. For these cases, you would use **zcat** or **grep** as filters.

Finally, you will have to enter the log files that are to be processed, either on a daily or weekly basis.

Installing the Cron Job

Installing the cron job is really easy: **lr_config** will give you the appropriate lines to add to you cron job. The lines to add to your crontab should look similar to :

```
0 10 * * * /usr/local/bin/lr_cron daily
0 10 * * 0 /usr/local/bin/lr_cron weekly
```

Once activated like this, report(s) will be sent on a weekly and/or daily basis.

Automatically Processing Log Files Through A Responder

Automatically Processing Log Files In A Server Farm

Chapter 5. Customizing Lire's Reports

This chapter explains how to customize the reports generated by Lire. Each superservice comes with a number of subreports which select various information from the log file for inclusion in the final report.

The report's configuration for a specific superservice is in a file named *`superservice`*`.cfg`. This file is looked for in `$sysconfdir/lire` and `$HOME/.lire/etc/`. If you're not happy with the default configuration as shipped with Lire, you're advised to copy `$sysconfdir/lire/`*`superservice`*`.cfg` to `$HOME/.lire/etc/` and edit this copy. This copy will of course not get touched on Lire upgrades.

The Report's Configuration File

The configuration file is *line oriented*. Empty lines are ignored as well as line starting with `#`. The configuration file is divided in sections introduced by lines starting with the directive `=section`. The section's title follows the directive.

Each section can optionally contains a list of filters that will be used to filter the records used to generate the subreports. A filter is setup by starting a line with the pipe (`|`) character followed by the filter's id. The rest of the line will usually contains filter's parameter assignments. If any filters is used in the section, they must come after the section's title and before the subreports.

Other lines are interpreted as subreport's id.

Example 5-1. Commented Report Configuration File

Here is an example configuration for the DNS superservice.

```
=section All Requests
top-requesting-hosts               hosts_to_show=10
top-requested-names                names_to_show=10
requesttype-distribution
requests-by-period                 period=1d

=section Recursive Requests
|select-resolver    method="recurs"
top-requesting-hosts               hosts_to_show=10
top-requested-names                names_to_show=10
requesttype-distribution
requests-by-period                 period=1d

=section Non Recursive Requests
|select-resolver    method="nonrec"
top-requesting-hosts               hosts_to_show=10
top-requested-names                names_to_show=10
requesttype-distribution
requests-by-period                 period=1d
```

This DNS report will contains three sections. The first section "All Requests" doesn't use any filters and thus the four configured subreports (`top-requesting-hosts`, `top-requested-names`, `requesttype-distribution` and `requests-by-period`) will be computed using all the requests.

The second section ("Recursive Requests") has one filter setup (`select-resolver`) which will selects only recursive requests. This section will contains the same subreports than the "All Requests" section, but calculated on a different input set. The third section ("Non Recursive Requests") is similar to the second section with the exception that only non-recursive requests will be used to compute the subreports.

Selecting Subreports

Example 5-2. FTP Report Configuration File

Here is an example configuration file for the FTP superservice.

```
# Report configuration for the FTP super service

# Top X reports
top-remote-host hosts_to_show=10
#top-files files_to_show=10
top-files-in files_to_show=10
top-files-out files_to_show=10
top-users users_to_show=10

# By day reports
bytes-by-period period="1d"

# Transfers by X reports
transfers-by-direction
transfers-by-type
```

The FTP superservice will thus contain seven subreports. Because the line with the `top-files` subreport starts with #, this subreport won't be included in the report. To include that subreport in the report, you would have to remove the # character.

In Part II, you will find all the subreports and filters available for all superservices.

Reordering The Subreports

Ordering is very simple. The order in which subreport lines appear in the config files, is the order in which the subreports will be given in the output. Rearranging the lines in these configuration files reorders them in the output. For example, in the above example, `transfers-by-type` will be the last report given in the output. You can reorder section in a similar manner.

Changing Parameters

Many subreports (and filters) can be customized through the configuration files. For example, consider this excerpt from the DNS configuration file previously listed.

```
top-requesting-hosts                hosts_to_show=10
top-requested-names                 names_to_show=10
requesttype-distribution
requests-by-period                  period=1d
```

All the reports are selected, but furthermore, for the subreports giving a "Top *X*" output the number *X* can be defined. With the above configuration the report `top-requesting-hosts` will give a Top 10.

> **Caution**
>
> All variable settings must be placed on the same line as the subreport's id or filter's id!

A more exotic example is taken from the WWW superservice configuration file:

```
top-referers-by-page referer_to_show=5 page_to_show=10 referer_exclusion='^-$'
```

In this example a Perl regular expression is used as content for the *`referer_exclusion`* variable. This expression matches all referers `-`. Such referers are found in the log file in cases when e.g. the URL of your web page was typed by the client user. (When users visit your page by clicking on a link in a page, refering to your page, the page linked from will be given in the referer field.) All referers that match `-` will be excluded from the analysis.

In Part II, you will find the description of all the available subreports along with their parameters.

Using Subreports On Filtered Input

Some subreport contains support for filtering their input. One such subreport is the `top-referers-by-page` subreport. But sometime a subreport doesn't support filtering its input, or you could want to have exactly the same information as other subreports already included in the report but on a subset of the records. The solution to this problem is to setup filters in a specific section. You can find an example of this in the DNS report's configuration file that was commented in the Section called *The Report's Configuration File*.

In Part II, you will find the description of all the available filters along with their parameters.

II. Reports Reference

Chapter 6. Database Reports

Supported Log Format

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 contains information about each the start and shutdown of your database server, as well as all connections and queries processed by the database server during its session.

Example 6-1. 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)
```

Reports' Descriptions and Configuration

Actions By Period Database Report

ID: `actions-by-period`
Chart: None

This report shows the number of actions in configurable time periods.

Parameters

`period`

This parameter controls the time period over which the deliveries are aggregated.

Defaults to `1d`.

Most Active Users Database Report

ID: `top-users`
Chart: bars

This report lists the users that do the most actions.

Parameters

`users_to_show`

This parameter controls the number of users to display in the report.

Defaults to `10`.

Most Accessed Databases Database Report

ID: `top-databases`
Chart: bars

This report lists the databases that do are most accessed.

Parameters

`databases_to_show`

This parameter controls the number of users to display in the report.

Defaults to `10`.

Queries By Type

ID: `top-querytypes`
Chart: bars

This report shows the number of queries by query type.

This report doesn’t have any parameters.

Filters’ Descriptions and Configuration

None.

Chapter 7. DNS Reports

Supported Log Format

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

Note: You have to enable query logging which isn't 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's 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 than Bind 8 (except wether the request was recursive or not) but in another format.

Example 7-3. Sample Bind 9's 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 miss 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
```

Reports' Descriptions and Configuration

Top Requesting Hosts Report

ID: `top-requesting-hosts`
Chart: bars

This report lists the requesting hosts with the most requests.

Parameters

`hosts_to_show`

This parameter controls the number of hosts to display in the report.

Defaults to `10`.

Top Requesting Hosts By Method Report

ID: `top-requesting-hosts-by-method`
Chart: bars

This report lists the requesting hosts with the most requests aggregated by method.

Parameters

`hosts_to_show`

This parameter controls the number of hosts to display in the report.

Defaults to `10`.

`method`

This parameter filters the requests just for one method. By defaults the requests are given for the "recurs" method. Possible methods are:

nonrec

Non-recursive requests.

recurs

Recursive requests.

Defaults to `^recurs$`.

Top Requested Names Report

ID: `top-requested-names`
Chart: bars

This report lists the most requested names.

Parameters

`names_to_show`

This parameter controls the number of names to display in the report.

Defaults to `10`.

Top Requested Names By Method Report

ID: `top-requested-names-by-method`
Chart: bars

This report lists the most requested names aggregated by method.

Parameters

`names_to_show`

This parameter controls the number of names to display in the report.

Defaults to `10`.

`method`

This parameter filters the requests just for one method. By defaults the requests are given for the "recurs" method. Possible methods are:

nonrec

Non-recursive requests.

recurs

Recursive requests.

Defaults to `^recurs$`.

Distribution of Request Types by Method DNS Report

ID: `requesttype-by-method`
Chart: None

This report shows the distribution of the type of requests splitted by method.

This report doesn't have any parameters.

Distribution of Request Types Report

ID: `requesttype-distribution`
Chart: bars

This reports on the distribution of the request types.

This report doesn't have any parameters.

Distribution of Request Types By Method Report

ID: `requesttype-distribution-by-method`
Chart: bars

This reports on the distribution of the request types aggregated over a method.

Parameters

`method`

This parameter filters the requests just for one method. By defaults the requests are given for the "recurs" method. Possible methods are:

nonrec

Non-recursive requests.

recurs

Recursive requests.

Defaults to `^recurs$`.

Requests Summary DNS Report

ID: `requests-summary`
Chart: None

This report shows some global statistics about the requests made on the DNS server.

This report doesn't have any parameters.

Requests Summary by Method DNS Report

ID: `requests-summary-by-method`
Chart: None

This report shows some global statistics about the requests made on the DNS server splitted by method.

This report doesn't have any parameters.

Requests By Period DNS Report

ID: `requests-by-period`
Chart: histogram

This report shows the number of requests aggregated in configurable time periods.

Parameters

`period`

This parameter controls the time period over which the requests are aggregated.

Defaults to `1d`.

Requests By Period By Method DNS Report

ID: `requests-by-period-by-method`
Chart: None

This report shows the number of requests aggregated in configurable time periods.

Parameters

`period`

This parameter controls the time period over which the requests are aggregated.

Defaults to `1d`.

`method`

This parameter filters the requests just for one method. By defaults the requests are given for the "recurs" method. Possible methods are:

nonrec

Non-recursive requests.

recurs

Recursive requests.

Defaults to `^recurs$`.

Requests By Timeslot DNS Report

ID: `requests-by-timeslot`
Chart: histogram

This report shows the number of requests aggregated in configurable time slots (by hours of the day, by days of the weeks, etc.).

Parameters

`timeslot`

This parameter controls the time slot over which the requests are aggregated.

Defaults to `1h`.

Requests by Period by Method DNS Report

ID: `req-by-period-by-method`
Chart: None

This report shows the number of requests aggregated in configurable time periods and splitted by the resolver’s type.

Parameters

`period`

This parameter controls the time period over which the requests are aggregated.

Defaults to `1d`.

Requests by Timeslot by Method DNS Report

ID: `req-by-timeslot-by-method`
Chart: None

This report shows the number of requests aggregated in configurable time slots (by hours of the day, by days of the weeks, etc.) and splitted by the resolver's type.

Parameters

`timeslot`

This parameter controls the time slot over which the requests are aggregated.

Defaults to `1h`.

Filters' Descriptions and Configuration

Select Resolver Filter

ID: `select-resolver`

This filter specification can be used to select only DLF records having a particular resolver type.

Parameters

`method`

This parameter sets the type of resolver that should be selected. Possible methods are:

nonrec

Non-recursive requests.

recurs

Recursive requests.

Defaults to `^recurs$`.

Chapter 8. Email Reports

Supported Log Format

Lire supports log files from four different email servers.

Exim

The standard log file from Exim are supported.

Example 8-1. 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 8-2. 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 8-3. 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 8-4. 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 8.11.x of Sendmail are supported.

Example 8-5. 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
```

Reports' Descriptions and Configuration

Deliveries Attempts By Period By Status Email Report

ID: `deliveries-by-period-by-status`
Chart: None

This report shows the number of delivery attempts that resulted in a specific status, aggregated in configurable time periods.

Parameters

`period`

This parameter controls the time period over which the deliveries are aggregated.

Defaults to `1d`.

Deliveries Attempts By Period Email Report

ID: `deliveries-by-period`
Chart: None

This report shows the number of delivery attempts aggregated in configurable time periods.

Parameters

`period`

This parameter controls the time period over which the deliveries are aggregated.

Defaults to `1d`.

Deliveries Attempts By Delay Email Report

ID: `deliveries-by-delay`
Chart: None

This report shows the number of deliveries attempts aggregated in configurable delay range.

Parameters

`delay-size`

This parameter controls the size of the delay ranges over which the deliveries are aggregated.

Defaults to `1s`.

Deliveries Attempts By Size Email Report

ID: `deliveries-by-size`
Chart: None

This report shows the number of deliveries attempts aggregated in configurable size range.

Parameters

`size`

This parameter controls the size of the size ranges over which the deliveries are aggregated.

Defaults to `1k`.

Failed Deliveries By Relay Email Report

ID: `errors-by-to-relay`
Chart: None

This report shows the errors that happened for each relay that was contacted.

This report doesn’t have any parameters.

Highest Average Delay By To Relay And To Domain Email Report

ID: `top-avg-delay-by-to-relay-and-to-domain`
Chart: None

This report shows the relay and domain with the highest average delay.

Parameters

delay_to_show

This parameter controls the number of delay to display in the report.

Defaults to `10`.

Most Deliveries Between Relays Email Report

ID: `top-deliveries-btw-relays`
Chart: bars

This report lists the connections between two relays with the most deliveries.

Parameters

connection_to_show

This parameter controls the number of connections to display in the report.

Defaults to `10`.

Most Deliveries From Domain Email Report

ID: `top-from-domain`
Chart: bars

This report lists the domain from which we received the most emails.

Parameters

domain_to_show

This parameter controls the number of domains to display in the report.

Defaults to `10`.

Most Deliveries From User By Domain Email Report

ID: `top-from-email-by-domain`
Chart: None

This report lists the user by domain from which we received the most emails.

Parameters

domain_to_show

This parameter controls the number of domains to display in the report.

Defaults to `30`.

user_to_show

This parameter controls the number of user by domain to display in the report.

Defaults to `5`.

Most Deliveries From Relay Email Report

ID: `top-from-relay`
Chart: bars

This report lists the relay from which we received the most emails.

Parameters

relay_to_show

This parameter controls the number of relays to display in the report.

Defaults to `10`.

Largest Email Exchange Email Report

ID: `top-largest-email-exchange`
Chart: None

This report the sender and recipient that exchange the largest volume of email.

Parameters

exchange_to_show

This parameter controls the number of sender, recipient to display in the report.

Defaults to `10`.

`msg_to_show`

This parameter controls the number of messages to display in the report.

Defaults to `5`.

Most Deliveries To Domain Email Report

ID: `top-to-domain`
Chart: bars

This report lists the domain to which we sent the most emails.

Parameters

`domain_to_show`

This parameter controls the number of domains to display in the report.

Defaults to `10`.

Most Deliveries To User By Domain Email Report

ID: `top-to-email-by-domain`
Chart: None

This report lists the user by domain to which we sent the most emails.

Parameters

`domain_to_show`

This parameter controls the number of domains to display in the report.

Defaults to `30`.

`user_to_show`

This parameter controls the number of user by domain to display in the report.

Defaults to `5`.

Most Deliveries From Relay Email Report

ID: `top-to-relay`
Chart: bars

This report lists the relay to which we sent the most emails.

Parameters

`relay_to_show`

This parameter controls the number of relays to display in the report.

Defaults to `10`.

Largest Volume Received From Domain Email Report

ID: `top-volume-from-domain`
Chart: bars

This report lists the domain from which we received the largest volume of mail.

Parameters

`domain_to_show`

This parameter controls the number of domains to display in the report.

Defaults to `10`.

Largest Volume Sent To Domain Email Report

ID: `top-volume-to-domain`
Chart: bars

This report lists the domains to which the largest volume of mail was sent.

Parameters

`domain_to_show`

This parameter controls the number of domains to display in the report.

Defaults to `10`.

Tracked Recipients Email Report

ID: `tracked-recipients`
Chart: None

This report shows the number of messages received from each sender for selected recipient emails.

Parameters

`tracked_email_re`

This parameter controls which recipient emails will be included in the report.

Defaults to `.*`.

Tracked Senders Email Report

ID: `tracked-senders`
Chart: None

This report shows the number of messages sent to each recipients for selected sender emails.

Parameters

`tracked_email_re`

This parameter controls which sender emails will be included in the report.

Defaults to `.*`.

Volume Delivered By Period Email Report

ID: `volume-by-period`
Chart: None

This report shows the volume of delivered emails in configurable time period.

Parameters

`period`

This parameter controls the time period over which the deliveries are aggregated.

Defaults to `1d`.

Filters’ Descriptions and Configuration

None.

Chapter 9. Firewall Reports

Supported Log Format

Lire supports logs from many packet filters 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 9-1. 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 through **syslog** (actually the kernel log buffer which is usually sent to **syslog**) packets marked for logging. Lire expects the logs in the form of a syslog log file.

Example 9-2. 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 9-3. 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 through **syslog** (actually the kernel log buffer which is usually sent to **syslog**) packets marked for logging. 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 9-4. 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**. Lire either native WELF log file or **syslog**'s log files contains WELF information. Altough that log format isn't designed for packet filter firewall (it can contains information from devices that does network intrusion or proxy services), Lire does it best to map this information to something that can be meaningful.

Example 9-5. 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 9-6. 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
```

Reports’ Descriptions and Configuration

Bytes by Period Firewall Report

ID: `bytes-by-period`
Chart: histogram

This report shows the number of bytes aggregated in configurable time periods.

Parameters

period

This parameter controls the time period over which the bytes are aggregated.

Defaults to `1d`.

Traffic's Volume by Rule Firewall Report

ID: `bytes-by-rule`
Chart: bars

This report shows the volume of data logged by each rule.

This report doesn't have any parameters.

Bytes by Timeslot Firewall Report

ID: `bytes-by-timeslot`
Chart: histogram

This report shows the volume of traffic distributed by timeslots (hours of the day, days of the week, etc.) that passed (or were denied) by your firewall.

Parameters

timeslot

This parameter controls the length of the timeslot over which the packets are aggregated. Use **`1h`** for "hours of the day" or **`1d`** for "days of the week".

Defaults to `1h`.

Top Bytes per From-IP Report

ID: `bytesperfrom`
Chart: bars

This report lists the IP addresses sending the highest data volume.

Parameters

ips_to_show

This parameter controls the number of sending IP adresses to display in the report.

Defaults to `10`.

Top Bytes per From-IP per Port Report

ID: `bytesperfromperport`
Chart: bars

This report lists the volume we were asked to receive per source IP per source port.

Parameters

`ips_to_show`

This parameter controls the number of sending IP adresses to display in the report.

Defaults to `10`.

`ports_to_show`

This parameter controls the number of source ports to display in the report.

Defaults to `10`.

Top Bytes per To-ip Report

ID: `bytesperto`
Chart: bars

This report lists the IP addresses for which we were asked to sent the highest data volume to.

Parameters

`ips_to_show`

This parameter controls the number of receiving IP adresses to display in the report.

Defaults to `10`.

Top Bytes per destination IP per Port Report

ID: `bytespertoperport`
Chart: bars

This report lists the volume were asked to receive per destination IP per port.

Parameters

ips_to_show

This parameter controls the number of receiving IP adresses to display in the report.

Defaults to `10`.

ports_to_show

This parameter controls the number of ports to display in the report.

Defaults to `10`.

Top blocked tcp packets per source IP per destination port Report

ID: `deniedtcpperport`
Chart: bars

This report lists the destination ports for which we blocked the highest tcp data volume, along with the sending ip adresses

Parameters

ips_to_show

This parameter controls the number of sending IP adresses to display in the report.

Defaults to `10`.

ports_to_show

This parameter controls the number of destination ports to display in the report.

Defaults to `10`.

Packets by Period Firewall Report

ID: `pkt-by-period`
Chart: histogram

This report shows the number of packets logged by the firewall aggregated in configurable time period.

Parameters

`period`

This parameter controls the time period over which the packets are aggregated.

Defaults to `1d`.

Packets by Rule Firewall Report

ID: `pkt-by-rule`
Chart: bars

This report shows the number of packets logged by the firewall for each rules.

This report doesn't have any parameters.

Packets by Timeslot Firewall Report

ID: `pkt-by-timeslot`
Chart: histogram

This report shows the number of packets distributed by timeslots (hours of the day, days of the week, etc.).

Parameters

`timeslot`

This parameter controls the length of the timeslot over which the packets are aggregated. Use **`1h`** for "hours of the day" or **`1d`** for "days of the week".

Defaults to `1h`.

Packet Summary Firewall Report

ID: `pkt-summary`
Chart: None

This report shows some general statistics about the number of packets logged by your firewall.

This report doesn't have any parameters.

Top Volume to Destination by Source Firewall Report

ID: `top-bytes-dst-by-src`
Chart: None

This report will show for a number of source IP addresses that sent the most volume of traffic, the list of destination (destination IP and destination port).

Parameters

`src_to_show`

This parameter controls the number of source IP adresses to display in the report.

Defaults to `15`.

`dst_to_show`

This parameter controls the number of destination (IP address and port) to display for each source IP.

Defaults to `20`.

Top Volume to Destination by Source Firewall Report

ID: `top-bytes-src-by-dst`
Chart: None

This report will show for each destination (destination IP and port) the list of source IPs that sent the most volume.

Parameters

`dst_to_show`

This parameter controls the number of destination (IP address and port) to display in the report.

Defaults to `15`.

`src_to_show`

This parameter controls the number of source IP adresses that will be displayed for each destination.

Defaults to `20`.

Top Messages Firewall Report

ID: `top-msg`
Chart: bars

This report shows the top messages (IDS alerts or others) generated by the firewall.

Parameters

`msgs_to_show`

This parameter controls the number of messages to show in the report.

Top Messages Firewall Report

ID: `top-dst-by-msg`
Chart: None

This report shows the top destination IPs that are the target of the messages (IDS alerts or others) generated by the firewall.

Parameters

`msgs_to_show`

This parameter controls the number of messages to show in the report.

`ips_to_show`

This parameter controls the number of destination IPS to list with each message.

Top Messages Firewall Report

ID: `top-src-by-msg`
Chart: None

This report shows the top source IPs that are at the origin of the messages (IDS alerts or others) generated by the firewall.

Parameters

`msgs_to_show`

This parameter controls the number of messages to show in the report.

`ips_to_show`

This parameter controls the number of source IPS to list with each message.

Top Packets by Source IP Report

ID: `top-pkt-by-src`
Chart: bars

This report lists the IP addresses that were listed as source in the most packets.

Parameters

`ips_to_show`

This parameter controls the number of source IP adresses to display in the report.

Defaults to `10`.

Top Packets by Destination IP Report

ID: `top-pkt-by-dst`
Chart: bars

This report lists the IP addresses that were listed as destination in the most packets.

Parameters

`ips_to_show`

This parameter controls the number of destination IP adresses to display in the report.

Defaults to `10`.

Top Packets Destination by Source Firewall Report

ID: `top-pkt-dst-by-src`
Chart: None

This report will show for a number of source IP addresses that sent the most packets, the list of destination (destination IP and destination port).

Parameters

`src_to_show`

This parameter controls the number of source IP adresses to display in the report.

Defaults to `15`.

dst_to_show

This parameter controls the number of destination (IP address and port) to display for each source IP.

Defaults to `20`.

Top Packets Source by Destination Firewall Report

ID: `top-pkt-src-by-dst`
Chart: None

This report will show for each destination (destination IP and port) the list of source IPs that sent the most packets.

Parameters

dst_to_show

This parameter controls the number of destination (IP address and port) to display in the report.

Defaults to `15`.

src_to_show

This parameter controls the number of source IP adresses that will be displayed for each destination.

Defaults to `20`.

Volume Summary Firewall Report

ID: `vol-summary`
Chart: None

This report shows some general statistics about the size of the packets logged by your firewall.

This report doesn't have any parameters.

Filters' Descriptions and Configuration

Select Action Filter

ID: `select-action`

This filter specification can be used to select only the firewall events that were permitted or denied.

Parameters

`action_match`

This parameter contains the action that should selected:

`denied`

Select only denied events.

`permitted`

Select only permitted events.

`-`

This is also a possible action when we can't determine from the log information if this event was `denied` or `permitted`.

Defaults to `denied`.

Chapter 10. FTP Reports

Supported Log Format

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 10-1. 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 a wide range of FTP servers like Wu-Ftpd, ProFTPD or standard BSD **ftpd**.

Example 10-2. 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
```

Reports' Descriptions and Configuration

Top Remote Host FTP Report

ID: `top-remote-host`
Chart: bars

This report lists the remote hosts with the most requests.

Parameters

`hosts_to_show`

This parameter controls the number of remote hosts to display in the report.

Defaults to `10`.

Bytes By Day FTP Report

ID: `bytes-by-day`
Chart: histogram

This report calculates the sum of all transfers by day.

This report specification is now obsolete and was replaced by the bytes-by-period report specification.

This report doesn't have any parameters.

Bytes by Period FTP Report

ID: `bytes-by-period`
Chart: histogram

This report calculates the sum of all transfers by a user-confirable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Bytes by User by Period FTP Report

ID: `bytes-by-user-by-period`
Chart: None

This report shows the bytes transferred by user by period.

Parameters

`period`

This parameter controls the time period which is used to aggregate the records.

Defaults to `1d`.

`users_to_show`

This parameter controls the number of users to display during each users.

Defaults to `10`.

Bytes by Direction by User with count by Period FTP Report

ID: `bytes-by-dir-by-user-by-period`
Chart: None

This report shows the bytes transferred by direction (in or out) by user with the number of transfers by period.

Parameters

period

This parameter controls the time period which is used to aggregate the records.

Defaults to `1d`.

users_to_show

This parameter controls the number of users to display during each period.

Defaults to `10`.

Top Files FTP Report

ID: `top-files`
Chart: bars

This report lists the most requested files.

Parameters

files_to_show

This parameter controls the number of files to display in the report.

Defaults to `10`.

Top Uploaded Files FTP Report

ID: `top-files-in`
Chart: bars

This report lists the most uploaded files.

Parameters

files_to_show

This parameter controls the number of files to display in the report.

Defaults to `10`.

Top Downloaded Files FTP Report

ID: `top-files-out`
Chart: bars

This report lists the most downloaded files.

Parameters

`files_to_show`

This parameter controls the number of files to display in the report.

Defaults to `10`.

Top Users FTP Report

ID: `top-users`
Chart: bars

This report lists the most active users.

Parameters

`users_to_show`

This parameter controls the number of users to display in the report.

Defaults to `10`.

Top by User (Bytes Transferred) FTP Report

ID: `top-users-bytes`
Chart: bars

This report lists the users with the highest amount of bytes transferred.

Parameters

`users_to_show`

This parameter controls the number of users to display in the report.

Defaults to `10`.

Tracked Users FTP Report

ID: `tracked-users`
Chart: None

This report shows each download for tracked users.

Parameters

`tracked_user_re`

This parameter controls which user will be included in the report.

Defaults to `.*`.

Tracked Files FTP Report

ID: `tracked-files`
Chart: None

This report shows each download of tracked files.

Parameters

`tracked_file_re`

This parameter controls which files will be included in the report.

Defaults to `.*`.

Number of Transfers by Direction FTP Report

ID: `transfers-by-direction`
Chart: pie

This report lists the number of transfers by direction.

This report doesn’t have any parameters.

Number of Transfers by Transfer Type FTP Report

ID: `transfers-by-type`
Chart: pie

This report lists the number of transfers by transfer type.

This report doesn’t have any parameters.

Filters’ Descriptions and Configuration

None.

Chapter 11. Print Reports

Supported Log Format

The print superservice supports printer log 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 11-1. 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 11-2. 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'
```

Reports' Descriptions and Configuration

Jobs per Printer Print Report

ID: `jobs-per-printer`
Chart: bars

This report shows the number of jobs for each printer.

This report doesn't have any parameters.

Top Users Print Report

ID: `top-users`
Chart: bars

This report lists the users with the most print jobs.

Parameters

`users_to_show`

This parameter controls the number of users to display in the report.

Defaults to `10`.

Jobs per Printer per Period Print Report

ID: `jobs-per-printer-per-period`
Chart: None

This report lists the daily total of jobs for each printer.

Parameters

`period`

This parameter controls the time period over which the jobs are aggregated.

Defaults to `1d`.

Filters’ Descriptions and Configuration

None.

Chapter 12. Proxy Reports

Supported Log Format

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

Microsoft Internet Security and Acceleration Server

That 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 12-1. 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's access log.

Example 12-2. 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**. Lire either native WELF log file or **syslog**'s log files contains WELF information. This format can be used by packet filters firewall, proxies or network intrusion detection devices. Lire will only process records that are related through proxy services (either application proxy like a web proxy or a transport proxy like for the telnet protocol).

Example 12-3. 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
```

Reports’ Descriptions and Configuration

Bytes by Cache Result

ID: `bytes-by-cache_result`
Chart: bars

This report shows the number of bytes transferred for each cache result. This gives an idea of how effective is the cache.

This report doesn’t have any parameters.

Bytes by Object’s Source

ID: `bytes-by-result_src_code`
Chart: bars

This report shows the number of bytes transferred from each source (Internet, Parent or sibling cache, etc.). This report is useful to analyze the performance of your array.

This report doesn’t have any parameters.

Bytes Transferred By Period Proxy Report

ID: `bytes-by-period`
Chart: histogram

This report shows the number of bytes transferred by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Bytes Transferred By Timeslot Proxy Report

ID: `bytes-by-timeslot`
Chart: histogram

This report shows the number of bytes transferred by user-configurable timeslot (hours of the day, days of the week, etc.). This report is useful to spot the most used period of the day for example.

Parameters

`timeslot`

This parameter controls the unit of time used to aggregate the records.

Defaults to `1h`.

Client Summary Proxy Report

ID: `clients-summary`
Chart: None

This report shows some global stats about the client hosts.

This report doesn't have any parameters.

Requests Summary Proxy Report

ID: `requests-summary`
Chart: None

This report shows some global stats about proxy requests.

This report doesn't have any parameters.

Requests by Cache Result

ID: `requests-by-cache_result`
Chart: bars

This report shows the number of requests for each cache result. This gives an idea of how effective is the cache.

This report doesn't have any parameters.

Requests By Period Proxy Report

ID: `requests-by-period`
Chart: histogram

This report shows the number of requests by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Requests By Size Proxy Report

ID: `requests-by-size`
Chart: histogram

This report shows the number of requests by size. It shows the number of requests that were between 1k-5k, 5k-10k, etc.

Parameters

`range_size`

This parameter controls the size of the first range class.

Defaults to `1k`.

Number of Requests By Timeslot Proxy Report

ID: `requests-by-timeslot`
Chart: histogram

This report shows the number of requests by user-configurable timeslot (hours of the day, days of the week, etc.).

Parameters

`timeslot`

This parameter controls the unit of time used to aggregate the records.

Defaults to `1h`.

Requests By Request's Time Proxy Report

ID: `requests-by-time`
Chart: histogram

This report shows the number of requests by the time they took to process. It shows the number of requests that took between 0-1s, 1-2s, etc.

Parameters

`range_size`

This parameter controls the size of the first range class.

Defaults to `1s`.

Top Clients by Destinations Proxy Report

ID: `top-clients-by-destinations`
Chart: None

This report lists the most popular destinations along with the clients that accessed them.

Parameters

`clients_to_show`

This parameter controls the number of clients for each destination to display in the report.

Defaults to `10`.

dsts_to_show

This parameter controls the number of destinations to display in the report.

Defaults to `10`.

Top Destinations by Number of Requests

ID: `top-destinations`
Chart: bars

This report lists most popular sites

Parameters

dsts_to_show

This parameter controls the number of destinations to display in the report.

Defaults to `10`.

Top Destinations by Bytes Downloaded

ID: `top-destinations-by-bytes`
Chart: bars

This report lists most popular sites by traffic

Parameters

dsts_to_show

This parameter controls the number of destinations to display in the report.

Defaults to `10`.

Top Destinations by Clients

ID: `top-destinations-by-clients`
Chart: None

This report lists most popular sites by clients

Parameters

dsts_to_show

This parameter controls the number of destinations to display in the report.

Defaults to `10`.

clients_to_show

This parameter controls the number of clients to display in the report.

Defaults to `10`.

Top Destinations by Users Proxy Report

ID: `top-destinations-by-users`
Chart: None

This report lists most popular destinations, grouped by users

Parameters

users_to_show

This parameter controls the number of users for each URL to display in the report.

Defaults to `10`.

dsts_to_show

This parameter controls the number of destinations to display in the report for each user.

Defaults to `10`.

Top Users by Destinations Proxy Report

ID: `top-users-by-destinations`
Chart: None

This report lists the most popular destinations along with the users that accessed them.

Parameters

users_to_show

This parameter controls the number of users for each destinations to display in the report.

Defaults to `10`.

dsts_to_show

This parameter controls the number of destinations to display in the report.

Defaults to `10`.

Top MIME types by Transferred Size

ID: `top-types-by-bytes`
Chart: bars

This report lists the MIME types that resulted in the most traffic.

Parameters

types_to_show

This parameter controls the number of MIME types to display in the report.

Defaults to `10`.

Top Users by Bytes Proxy Report

ID: `top-users-by-bytes`
Chart: bars

This report lists the users who downloaded biggest volume using the proxy

Parameters

users_to_show

This parameter controls the number of users to display in the report.

Defaults to `10`.

Top URLs by Users Proxy Report

ID: `top-urls-by-users`
Chart: None

This report lists most popular URLs, grouped by users

Parameters

`users_to_show`

This parameter controls the number of users for each URL to display in the report.

Defaults to `10`.

`urls_to_show`

This parameter controls the number of URLs to display in the report.

Defaults to `10`.

User Summary Proxy Report

ID: `users-summary`
Chart: None

This report shows some global stats about the users.

This report doesn't have any parameters.

Filters' Descriptions and Configuration

Select Cache Result Filter

ID: `select-cache_result`

This filter specification can be used to select only DLF records in the proxy superservice which have a particular cache_result value. For example, you could select only denied requests by using the value `TCP_DENIED`.

Parameters

`result`

This parameter is a regular expression which will be used to select the codes you want.

Defaults to `TCP_DENIED`.

Chapter 13. WWW Reports

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 never used.

authuser

The authenticated username.

date

The timestamp of the request.

request

The first line of the 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 webs 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 request's header. This usually reflects the page the user visited before this request.

useragent

The content of the User-Agent request's header. 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 the 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 informations 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 informations 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 request's header 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
```

Altough 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" www common joe@example.com < $f; done
```

W3C Extended Log Format

This is a log format defined by the W3C which can contain variable 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.

Reports' Descriptions and Configuration

Bytes By Day WWW Report

ID: `bytes-by-day`
Chart: histogram

This report calculates the sum of all transfers by day.

This report specification is now obsolete and was replaced by the bytes-by-period report specification.

This report doesn't have any parameters.

Bytes By Period WWW Report

ID: `bytes-by-period`
Chart: histogram

This report calculates the sum of all transfers by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the requests' size.

Defaults to `1d`.

Bytes Per Directory WWW Report

ID: `bytes-by-dir`
Chart: bars

This report shows the amount of data requested by directory. This amount only includes the size of request in that directory, not of any child directories.

Parameters

`bytesdir_to_show`

This parameter controls the number of different directories to display in the report.

Defaults to `10`.

Bytes By HTTP Result By Day WWW Report

ID: `bytes-by-result-by-day`
Chart: None

This report calculates the size of all requests by HTTP result by day.

This report specification is now obsolete and was replaced by the bytes-by-result-by-period report specification.

This report doesn't have any parameters.

Bytes By HTTP Result By Period WWW Report

ID: `bytes-by-result-by-period`
Chart: None

This report calculates the size of all requests by HTTP result by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Bytes By HTTP Result WWW Report

ID: `bytes-by-result`
Chart: pie

This report calculates the size of all requests by HTTP result.

This report doesn't have any parameters.

Client Hosts by Day WWW Report

ID: `clienthost-by-day`
Chart: histogram

This report count the number of different hosts that made requests by day.

This report specification is now obsolete and was replaced by the clienthost-by-period report specification.

This report doesn't have any parameters.

Client Hosts By Period WWW Report

ID: `clienthost-by-period`
Chart: histogram

This report count the number of different hosts that made requests by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Requests By Browser WWW Report

ID: `requests-by-browser`
Chart: bars

This report shows the number of requests for each browser.

Parameters

`browsers_to_show`

This parameter controls the number of Browsers to display in the report.

Defaults to `10`.

Number of Requests By Day WWW Report

ID: `requests-by-day`
Chart: histogram

This report shows the number of requests by day.

This report specification is now obsolete and was replaced by the requests-by-period report specification.

This report doesn't have any parameters.

Number of Requests By Period WWW Report

ID: `requests-by-period`
Chart: histogram

This report shows the number of requests by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Requests By Browser Language WWW Report

ID: `requests-by-lang`
Chart: bars

This report shows the number of requests for each browser language.

Parameters

`lang_to_show`

This parameter controls the number of Browsers to display in the report.

Defaults to `10`.

Requests By HTTP Method WWW Report

ID: `requests-by-method`
Chart: pie

This report shows the number of requests for each HTTP method (POST, GET, PUT, etc.)

This report doesn't have any parameters.

Requests By OS WWW Report

ID: `requests-by-os`
Chart: bars

This report shows the number of requests for each browser OS.

Parameters

`os_to_show`

This parameter controls the number of OS to display in the report.

Defaults to `10`.

Requests By Result By Day WWW Report

ID: `requests-by-result-by-day`
Chart: None

This report calculates the number of requests by HTTP result by day.

This report specification is now obsolete and was replaced by the requests-by-period report specification.

This report doesn't have any parameters.

Requests By Result By Period WWW Report

ID: `requests-by-result-by-period`
Chart: None

This report calculates the number of requests by HTTP result by a user-configurable time period.

Parameters

`period`

This parameter controls the time period which is use to aggregate the records.

Defaults to `1d`.

Requests By HTTP Result WWW Report

ID: `requests-by-result`
Chart: pie

This report shows the number of requests for each HTTP result.

This report doesn't have any parameters.

Requests By Gzip Result WWW Report

ID: `requests-by-gzip_result`
Chart: pie

This report shows the number of requests for each Gzip result.

This report doesn't have any parameters.

Requests By Robot Report

ID: `requests-by-robot`
Chart: bars

This report shows the number of requests for each robot.

Parameters

`robots_to_show`

This parameter controls the number of Browsers to display in the report.

Defaults to `10`.

Requests By Top Level Domain Report

ID: `requests-by-country`
Chart: pie

This report shows the number of requests for top level domain.

This report doesn't have any parameters.

Requests By Attack Report

ID: `requests-by-attack`
Chart: bars

This report shows the number of requests for each attack.

This report doesn't have any parameters.

Requests By Keywords Report

ID: `requests-by-keywords`
Chart: bars

This report shows the number of requests that resulted from keywords in search engine.

This report doesn't have any parameters.

Requests By User Agent WWW Report

ID: `requests-by-useragent`
Chart: bars

This report shows the number of requests for each user agent.

This report doesn't have any parameters.

Number of Requests By Size WWW Report

ID: `requests-by-size`
Chart: histogram

This report shows the number of requests by request's size.

Parameters

`range_size`

This parameter controls the size of the first range class.

Defaults to `1k`.

Number of Requests By Timeslot WWW Report

ID: `requests-by-timeslot`
Chart: histogram

This report shows the number of requests by timeslot.

Parameters

timeslot

This parameter controls the unit of time used to aggregate the records.

Defaults to `1h`.

Requests By HTTP Protocol Version WWW Report

ID: `requests-by-version`
Chart: pie

This report shows the number of requests for each HTTP Protocol Version.

This report doesn't have any parameters.

Requests Summary WWW Report

ID: `requests-summary`
Chart: None

This report shows some global stats about requests.

This report doesn't have any parameters.

Average Compression By File Type WWW Report

ID: `top-avg-compression-by-file-type`
Chart: bars

This report lists show the file extension which are on average most compressed.

Parameters

file_type_to_show

This parameter controls the number of different extension to display in the report.

Defaults to `10`.

Most Averaged Compressed Requested File WWW Report

ID: `top-avg-compression-by-url`
Chart: bars

This report lists show the files with the most compression.

Parameters

`url_to_show`

This parameter controls the number of URL to display in the report.

Defaults to `10`.

Top Client By HTTP Result WWW Report

ID: `top-client_host-by-result`
Chart: None

This report lists the client hosts with the most requests for each HTTP results.

Parameters

`client_to_show`

This parameter controls the number of client hosts to display in the report.

Defaults to `5`.

Top Client WWW Report

ID: `top-client_host`
Chart: bars

This report lists the client hosts with the most requests.

Parameters

`client_to_show`

This parameter controls the number of client hosts to display in the report.

Defaults to `10`.

Last Pages By Session WWW Report

ID: `top-last_page`
Chart: None

This report shows the requested pages (this excludes pictures) which were the last page to be visited in a user session.

Parameters

`page_to_show`

This parameter controls the number of last pages to display.

Defaults to `10`.

`not_page_re`

This parameter contains a regular expression which is used to filter out images.

Defaults to `\.(png|gif|jpg)$`.

First Pages By Session WWW Report

ID: `top-first_page`
Chart: None

This report shows the requested pages (this excludes pictures) which were the first page to be visited in a user session.

Parameters

`page_to_show`

This parameter controls the number of last pages to display.

Defaults to `10`.

`not_page_re`

This parameter contains a regular expression which is used to filter out images.

Defaults to `\.(png|gif|jpg)$`.

Most Requested Pages By Client Host WWW Report

ID: `top-pages-by-client_host`
Chart: None

This report shows the most requested pages (this excludes pictures) by client host.

Parameters

`page_to_show`

This parameter controls the number of pages to display for each client host.

Defaults to `5`.

`client_to_show`

This parameter controls the number of client hosts to display.

Defaults to `10`.

`not_page_re`

This parameter contains a regular expression which is used to filter out images.

Defaults to `^(png|gif|jpg)$`.

Most Travelled Referer -> Page Connections WWW Report

ID: `top-referer-page-connections`
Chart: bars

This report lists the connections between a referers and a pages with the most requests.

Parameters

`connection_to_show`

This parameter controls the number of connections to display in the report.

Defaults to `10`.

`referer_exclusion`

This parameter can be used to exclude referers from the report. For example, it can be used to exclude internal links by setting it to "^http://www.yourdomain.com/$". It defaults to "^-$" which excludes all records for which the referer wasn’t specified.

Defaults to `^-$`.

`target_exclusion`

This parameter contains a regular expression which is used to filter out images as targets.

Defaults to `\.(png|gif|jpg)$`.

Top Referring Pages By Requested Page WWW Report

ID: `top-referers-by-page`
Chart: None

This report shows the top referring pages for the top requested page.

Parameters

`page_to_show`

This parameter controls the number of requested pages to display.

Defaults to `10`.

`referer_to_show`

This parameter controls the number of referring pages to show for each requested page.

Defaults to `5`.

`not_page_re`

This parameter contains a regular expression which is used to filter out images.

Defaults to `^(png|gif|jpg)$`.

`referer_exclusion`

This parameter can be used to exclude referers from the report. For example, it can be used to exclude internal links by setting it to "^http://www.yourdomain.com/$". It defaults to "^-$" which excludes all records for which the referer wasn't specified.

Defaults to `^-$`.

Most Requested Pages WWW Report

ID: `top-requests`
Chart: bars

This report lists the most requested pages.

Parameters

`page_to_show`

This parameter controls the number of pages to display in the report.

Defaults to `10`.

Most Requested Tracked Pages By Client Host WWW Report

ID: `top-tracked_pages-by-client_host`
Chart: None

This report shows the client hosts that have the most requests for "tracked" pages.

Parameters

`page_to_show`

This parameter controls the maximum number of tracked pages to display for each client host.

Defaults to `5`.

`client_to_show`

This parameter controls the number of client hosts to display.

Defaults to `10`.

`tracked_page_re`

This parameter contains a regular expression which is used to determine which pages are tracked..

Requested Tracked Pages By Period WWW Report

ID: `tracked_pages-by-period`
Chart: histogram

This report shows highest tracked page by time period.

Parameters

period

This parameter controls the time period over which the requests are aggregated.

Defaults to `1d`.

tracked_page_re

This parameter contains a regular expression which is used to determine which pages are tracked.

Most Requested URLs By Client Host WWW Report

ID: `top-urls-by-client_host`
Chart: None

This report shows the most requested URLs by client host.

Parameters

url_to_show

This parameter controls the number of URLs to display for each client host.

Defaults to `5`.

client_to_show

This parameter controls the number of client host to display.

Defaults to `10`.

User Sessions By Period WWW Report

ID: `user_sessions-by-period`
Chart: histogram

This report shows the number of user sessions by time period.

Parameters

period

This parameter controls the time period over which the user sessions are aggregated.

Defaults to `1d`.

Finished and Unfinished Session WWW Report

ID: `user_session-finished`
Chart: None

This report gives the numbers of finished and unfinished user sessions.

This report doesn't have any parameters.

Visit times User Session WWW Report

ID: `user_session-visit-times`
Chart: histogram

This report shows the time a user took to visit the website.

Parameters

`range_size`

This parameter controls the time period over which the visit times are aggregated.

Defaults to `1m`.

Page Counts User Session WWW Report

ID: `user_session-page_counts`
Chart: histogram

This report shows the number of pages for the visits.

Parameters

`range_size`

This parameter controls the ranges over which the number of pages per visit are aggregated.

Defaults to `1`.

Filters' Descriptions and Configuration

Select URL Filter

ID: `select-url`

This filter specification can be used to select only the requests for particular URL.

For example, this filter could be used to create subreports about downloadable files from your website.

Parameters

`url_match`

This parameter contains the regular expression that will be used to select the requests. Only requests where URL matches that regexp will be included in the subreports.

Defaults to `\.tar.gz$`.

Select Client Host Filter

ID: `select-client_host`

This filter specification can be used to select only the requests by a particular client.

Parameters

`client_match`

This parameter contains the regular expression that will be used to select the client. Only requests made by a client host matching that regexp will be included in the subreports.

Defaults to `.*`.

Exclude URL Filter

ID: `exclude-url`

This filter specification can be used to exclude requests for particular URLs.

For example, this filter could be use to create subreports excluding images or other multimedia file.

Parameters

`url_match`

This parameter contains the regular expression that will be used to filter out the requests. Requests made for a URL matching that regexp will be excluded from the subreports.

Defaults to `\.(png|jpg|gif|jpeg)$`.

Exclude Client Host Filter

ID: `exclude-client_host`

This filter specification can be used to exclude requests coming from particular hosts from the repots.

Parameters

`client_match`

This parameter contains the regular expression that will be used to filter out the client. Requests made by a client host matching that regexp will be excluded from the subreports.

Defaults to `.*`.

Exclude Referer Filter

ID: `exclude-referer`

This filter specification can be used to exclude requests with a particular referrer.

For example, this filter could be used to exclude internal referral from your subreports.

Parameters

`referer_match`

This parameter contains the regular expression that will be used to filter out the referrer. Requests for which the referer field matches that regexp will be excluded from the subreports.

Defaults to `^-$`.

III. Lire Reference

Chapter 14. 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 of this one 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-perl5archlibdir`

This option specifies where architecture dependent perl modules (like XML::Parser) will be installed.

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

`--with-spooldir`

This option specifies the default value of `LR_SPOOLDIR` which is the spool directory used by the responder (see the Section called *Responder Configuration Parameters* in Chapter 15 for a description of this variable). Unless you're running your own responder, this variable is not interesting.

`--with-archivedir`

When you're archiving your reports and logs using the archive feature (see the Section called *The Lire Archive and Temporary Files* in Chapter 15), this sets the default value of `LR_ARCHIVEDIR` (see the Section called *General Configuration Parameters* in Chapter 15 for a description of this variable.)

`--with-sgmldir`

Sets the directory where SGML files will looked for. The configuration script will look in some "standard" locations under that directory to find the DocBook DTD and stylesheets.

Per default, ./configure will look in`/usr/lib/sgml`, `/usr/share/sgml`, `/usr/local/lib/sgml` and `/usr/local/share/sgml`.

Installation Environment Variables

Some environment variables can be set prior to running the **./configure** to tune the installation process. This can be used to specify the location 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.

`PATHTOPERL`

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

`PATHTOJADE`

Sets the path to the **jade** DSSSL interpreter.

`PATHTOPDFJADETEX`

Sets the path to the **pdfjatetex** command.

`PATHTOXSLTPROC`

Sets the path to the **xsltproc** XSLT processor.

`DBK_XML_DTD`

Sets the path to the DocBook XML Document Type Declaration. This should point to the XML V4.1.2 DTD.

`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`.)

`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`.)

Chapter 15. Configuration Parameters

Configuration variables are set in `${prefix}/etc/lire/defaults`. If you want to override any of these variables, create a file `${prefix}/etc/lire/defaults.local`. This will not be touched when upgrading the system.

If you don't like fiddling with configuration files manually, you can use the interactive **lr_config** script to set things up for you. Just start that script now, and use the rest of this section as a reference when needed.

Users can create their own `~/.lire/etc/defaults` file. Settings in this file get added to (and possibly override) the ones in `${prefix}/lire/defaults`. Users might like to set e.g.

```
TMPDIR=$HOME/lire/tmp/.
```

(after creating this directory) in their `~/.lire/etc/defaults` files.

General Configuration Parameters

These are the parameters you most likely want to modify:

`DEFAULT_OUTPUT_FORMAT`

This sets the report's default output format. It can be one of `txt`, `xml`, `html`, `xhtml`, `pdf`, `logml`, `rtf` or `docbookx`.

`LR_SCALE_BYTES`

This parameter controls whether bytes values will get scaled to a more human readable format (`22.1M`, `5.0k`, etc.) in reports.

`LR_SCALE_SEC`

This parameter controls whether duration values will get scaled to a more human readable format (`1h`, `23.1m`, etc.) in reports.

`LR_SCALE_NUMBER`

This parameter controls whether number values will get scaled to a more human readable format (`1M`, `3.4k`, etc.) in reports.

`LR_TARGET_USER`

This sets the report's reader profile. Valid values are `sysadmin` or `manager`. Lire will tune the subreports' descriptions for that class of reader.

`LR_USERLEVEL`

This sets the amount of details you will get in subreport's description. Valid values are `normal` or `advanced`.

`INCLUDEIMAGES`

When this variable is set to `1`, Lire will include charts whenever possible in the generated report.

`LR_MAX_MEMORY`

This sets the log file's size threshold above which Lire will use an algorithm which is slower but takes less memory to generate the reports. For optimal performance, you should set this to half your available physical RAM (or more if you don't need a lot of RAM for normal system operation). It defaults to `40Megs`.

`ARCHIVE`

This variable controls whether files are archived. Files which are candidates for archiving are moved from `$TMPDIR` to the archive. Furthermore, metainfo about the archived files gets stored in a Lire database.

`LR_ARCHIVEDIR`

This variable sets the directory where files which are canditates for archiving will be archived.

Defaults to *`${prefix}`*`/var/lib/lire`.

Responder Configuration Parameters

When running a responder, the following variables should be set:

`LR_SPOOLINTERVAL`

This sets the amount of time that the responder will sleep between each spool run.

`LR_SPOOLDIR`

This sets spool directory that will be used by the responder. All requests to the online responder should be spooled to maildir format mailboxes under that directory.

`DISCLAIMERFILE`

File containing the disclaimer's text included with the mail sent by **lr_log2mail**.

`SIGNATUREFILE`

File containing the signature that will be added to the mail sent by **lr_log2mail**. Defaults to `${prefix}/etc/lire/signature`.

`REPLYTO`

Sets the reply-to address for mail sent by **lr_log2mail**.

`FROM`

Sets the from address for mail sent by **lr_log2mail**.

Your own `defaults.local` file could look like e.g.

```
KEEP=1
LR_SPOOLINTERVAL=10
TMPDIR=/home/hibou/logreport/tmp
DISCLAIMERFILE=${prefix}/etc/lire/disclaimer.local
FROM='MyDomain Lire Responder <log@mydomain.com>'
REPLYTO='MyDomain Support <support@mydomain.com>'
```

`disclaimer.local` could read:

```
The Online Report Responder Service is free of charge. We do not accept any
liability however incurred in respect of a failure to perform as is imputable
to the same for any loss direct or indirect.
```

Miscellaneous Configuration Parameters

Other variables that you may want to set:

`DEBUG`

When this variable is set, all of Lire's messages at loglevel info or debug will be output on STDERR. (They might end up in your syslog though, if you use **lr_run** to redirect these messages.)

`LR_KEEP_TEMP_DLF`

When this variable is set, the temporary DLF files created for extended and derived schema will be kept. This is only used to debug the extended or derived schema creators.

`KEEP`

Set this to 1 if you want to keep intermediate files in `TMPDIR`, e.g. for debugging.

`XSLT_PROCESSOR`

This sets the XSLT processor to use. Valid values are `xsltproc` or `none`. When none is used, a builtin convertor for XML is used to generate the ASCII reports. Other report format won't be available in this case. (xalan-c or sablotron are currenlty not supported.)

`TMPDIR`

Sets the directory where log files which are getting processed and intermediate files are stored.

Defaults to *`$HOME`*`/tmp`.

`LOGGING`

Sets where log messages should be sent. Valid values are `syslog` or `stderr`.

`FACILITY`

When `LOGGING` is set to `syslog`, this sets the **syslog** facility that will be used.

`LOGGERTAG`

Tag that will be used for **syslog** logging. Defaults to `lire`.

There are other variables, e.g. to overwrite the location for XML related tools and files. You can find the names, along with descriptive comments, in the `${prefix}/etc/lire/defaults` file.

The Lire Archive and Temporary Files

By default, files in intermediate formats are stored in the directory specified in the `TMPDIR` configurable variable, and removed after processing. If you set the `KEEP` configurable variable, those files won't be removed.

There is no rotate-like mechanism yet. If your `KEEP` is set to 1, you'll have to clean up `TMPDIR` manually once in a while.

Note: You could create a crontab entry like this :

```
0 3 * * 0 find $HOME/tmp -type f -ctime +14 | xargs rm
```

to clean up the all files older than two weeks. Run such an entry as a trusted user only.

Files which could be reused later (e.g. reports which could be merged (automatically merging will get supported in a later Lire release)) can be stored in an archive. If you like to build such an archive, set `ARCHIVE`: `ARCHIVE` indicates whether files should get archived. If set, files which are candidates for archiving are moved from `TMPDIR` to the archive. I.e.:

Table 15-1. To KEEP or to ARCHIVE?

file is candidate for archive	variable KEEP is	variable ARCHIVE is	file is kept in
yes	set	set	archive

file is candidate for archive	variable KEEP is	variable ARCHIVE is	file is kept in
yes	set	unset	TMPDIR
yes	unset	set	archive
yes	unset	unset	/dev/null
no	set	set	TMPDIR
no	set	unset	TMPDIR
no	unset	set	/dev/null
no	unset	unset	/dev/null

When ARCHIVE is set, a Lire database gets build in `$LR_ARCHIVEDIR/meta/index` (LR_ARCHIVEDIR is `/usr/local/var/lib/lire/data` per default). This database is used to keep metainformation on the archived files.

Chapter 16. Lire Logging and Error Messages

Logging

Lire can log its messages, and output it to either standard error (stderr) or to syslog using the **logger** program. Choosing between either one of them is done with the `LOGGING` variable in `*${sysconfdir}*/lire/defaults` or `~/.lire/etc/defaults`. (See also **lr_run**.)

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
```

Chapter 17. Lire Installation Layout

Service specific scripts should reside in *`libexecdir/service`*. Configuration in *`sysconfdir/service`*.