

LogReport's Lire: Integrated Analysis of all your Internet Services' Log Files

Joost van Baal joostvb@logreport.org

11 October 2002

Contents

Introduction

Slide 1

LogReport's Lire: Integrated Analysis of all your Internet Services' Log Files

Joost van Baal joostvb@logreport.org
LogReport <http://www.logreport.org/>

This paper gives an introduction to Lire, LogReport's tool for performing an integrated analysis of all ones Internet Services.

It is based upon presentations given at FOSDEM, the Free and Open Source Software Developers Meeting, Februari 2002 in Brussels, Belgium, and at ne2000, the Open Air Networking Event, July 2002, Nuenen, The Netherlands.

Joost van Baal and Wessel Dankers are employed as software developers by the Stichting LogReport Foundation; together with two other developers they work on maintaining and promoting Lire, LogReport's flagship product.

Next to working for LogReport, Joost van Baal is doing volunteer work for the Debian project. His main interests are programming in Perl and deploying Internet services on Unix platforms.

Wessel Dankers is a Computer Science student at the Universiteit Utrecht, next to this he his involved in several Free Software development projects.

1 Log file analysis

Log files are often treated like a wasteful by-product of IT activity: they sit somewhere in a dark corner of a computer system and are only examined occasionally, usually in case of after-the-fact reactive problem solving. The infamous *rotate* is the only application dealing with them. This is unfortunate. Log files contain the traces of computer activity, and by intelligently analyzing these traces one can learn a lot about the behavior of a system and its users.

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. When such tools exist, they are specific to one product, which means that you can't compare your qmail and Exim mail servers.

The Stichting LogReport Foundation detected this flaw in system administration and chose to serve a dual purpose: developing and maintaining Lire, our Open Source reporting and analysis software, and serving as a nexus of documentation, ideas, and thought on the topic of log files and their potential applications.

Slide 2

Table of Contents

- Log file analysis
- Lire Overview
- Lire's Architecture
- Lire's Future
- The LogReport Project
- More information, contact, questions

This talk will discuss the technical aspects of Lire as well as the organisational aspects of LogReport as an Open Source project.

Slide 3

Log file analysis

Log file analysis is

- too often neglected, but
- giving access to invaluable information; however
- tedious and time-consuming, so
- in need for both flexible and generic software.

2 Lire Overview

2.1 Lire benefits

Slide 4

Why use Lire?

Lire is

- generic
- flexible
- free, in both senses of the word
- actively maintained, in an open environment
- highly configurable
- very portable
- secure
- commercially supported

The LogReport project tries to tackle the problems as outlined above by developing Lire. Lire is a software package to generate useful reports from raw log files of various network programs.

Lire is flexible. The tool can be accessed via a command line interface, but can also be run from cron, and can even get accessed via an email interface.

Lire is Free Software. When using Lire, you'll have all the benefits of Open Source software. Lire is available at no cost, from our website on <http://www.logreport.org/>, and is licenced under the GNU General Public License.

Lire is actively being maintained by the LogReport team, which currently consists of five experienced software developers. The development can be followed live on our CVS on SourceForge. A new release gets shipped almost monthly.

Lire is highly configurable. All configuration files are in a very simple syntax. Of course, a userfriendly interface to write the configuration is shipped with Lire.

Lire is very portable. It runs on at least four different Unixen, GNU/Linux included. Since it's written in Perl, porting to different platforms is easy.

Lire is secure. It is run under a dedicated user account. No processes running as root are involved. Care is taken when installing Lire as an online responder. (Of course, this does not exempt the system administrator from defining and implementing her own security measures.)

Lire is commercially supported: the LogReport team offers commercial consultancy and tailor-made extensions on demand.

2.2 Which problems does Lire solve?

It enables one to schedule hardware upgrades, detect anomalies in usage from services. It can be used as a tool in building a traffic-based accounting system for external customers. It gives insight in who's talking to who, which is valuable for marketing and business planners.

Slide 5

Lire's users

Lire is valuable for both

- system administrators, and
- business managers

2.3 How does Lire do this?

Lire converts stuff like

```
1.example.com - - [03/Feb/2002:06:25:27 +0100] "GET /contact/lists/commit/msg01057.php HTTP/1.0" 200 11193 "-" "Googlebot/2.1 (+http://www.googlebot.com/bot.html)"
2.example.com - - [03/Feb/2002:06:25:46 +0100] "GET /robots.txt HTTP/1.0" 404 5126 "-" "htdig/3.1.5 (webmaster@logreport.org)"
2.example.com - - [03/Feb/2002:06:25:46 +0100] "GET / HTTP/1.0" 200 12745 "-" "htdig/3.1.5 (webmaster@logreport.org)"
1157.example.com - - [10/Feb/2002:05:23:38 +0100] "GET /css.php HTTP/1.1" 200 3682 "http://logreport.org/doc/gen/dns/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; Q312461)"
1158.example.com - - [10/Feb/2002:06:22:44 +0100] "GET /lire/ex/plain.php HTTP/1.1" 200 14253 "http://www.google.com/search?hl=en&q=ascii+text+pics&spell=1" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; Win 9x 4.90)"
```

to a graph like the one below, in Figure 1.

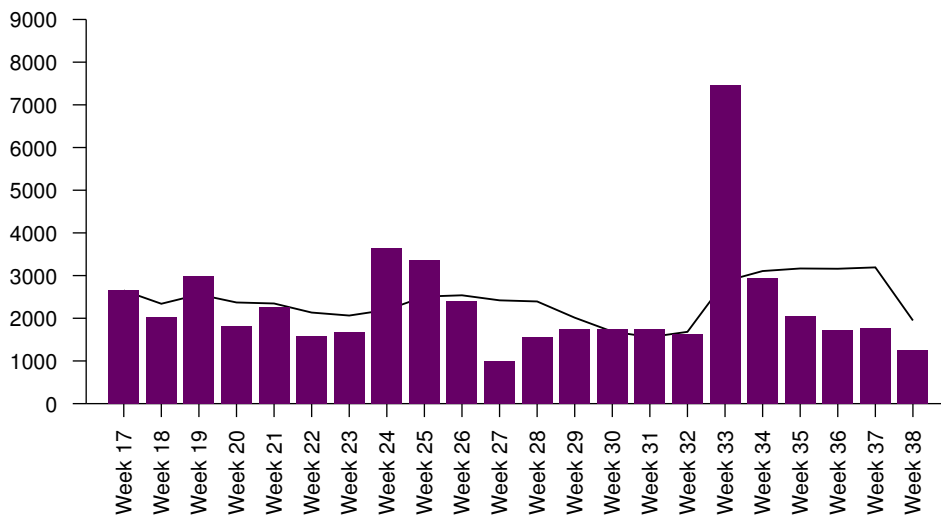


Figure 1: Lire tarball downloads from LogReport webserver, per week, May - Sept 2002

2.4 Lire supported log files and output formats

Slide 6

Lire supported log files

Lire currently supports log files from

- www and proxy (apache, IIS, squid, WELF, MS ISA)
- dns and dnszone (bind query and named logs)
- firewall (cisco IOS and PIX, Linux, IP Filter ...)
- email and msgstore (Exim, Postfix, qmail, sendmail, NMS, various POP and IMAP servers)
- ftp (ProFTPD, WU-FTPD, MS IIS)
- print (CUPS, LPRng)
- database (MySQL, PostgreSQL)
- dialup (isdn4linux) and syslog (generic syslog parser)

Multiple programs are supported for various types of network services:

- *www* log files in various formats from various webservers (Apache, IIS, Boa);
- *proxy* log files from squid and from WELF proxies, as well as from MS ISA servers;
- Bind version 8 and version 9 *dns* query logs and named logs;
- logs from *firewalls* such as IOS and PIX CISCO router, Linux ipchains, ipfilter and iptables, BSD IP Filter, WatchGuard, as well as logs in the WELF format as used by a lot of commercial firewall products;
- *email* logfiles from Exim, Postfix, qmail, sendmail, ArgoSoft and Netscape Messaging Server;
- POP and IMAP logs from various Netscape *msgstore* servers, as well as the DBMAIL server.
- log files from *ftp* servers in the xferlog format, as used by e.g. ProFTPD and WU-FTPD as well as logs from the MS IIS ftpserver;
- *print* logfiles (CUPS and LPRng);
- *database* transaction log files from MySQL servers;
- *dialup* logs from isdn4linux's isdnlog tool;
- *syslog* log files, with any mix of services.

Lire also supports various output formats for the generated reports: HTML, XHTML, XML, PDF, Excel 95, and plain ascii. Some of these formats support graphical representation of the data.

2.5 Lire installation

A Lire tarball is available for download from the LogReport website at <http://www.logreport.org/pub/>. A binary package for Debian GNU/Linux as well as an RPM is also available. Furthermore Lire is shipped with the FreeBSD ports collection.

Lire is written in Perl and shell code. Supported platforms are GNU/Linux, the BSD's and Sun Solaris, but since it's written in Perl, it very likely runs fine on a lot of other platforms too.

Slide 7

Ease of installation

Lire comes as a tarball (“autoconfiscated”), as an RPM and as a Debian package. A FreeBSD package is shipped with the FreeBSD ports collection. Written in Perl and shell, so runs on any Unix-like OS.

3 Lire’s Architecture

Slide 8

Table of Contents

- Log file analysis
- Lire Overview
- Lire’s Architecture
- Lire’s Future
- The LogReport Project
- More information, contact, questions

3.1 Overview

Internally, Lire represents the log file in a DLF file (for Distilled Log Format). This is a simple space-separated line-oriented ascii file. Each logged event is represented by one fixed-fields line.

A service coincides with one well-defined log file format. So, a service generally coincides with one application: the `sendmail` service handles sendmail log files. However, a lot of webservers use W3C defined formats, and a lot of commercial firewalls use the WELF format. Therefore, `w3c_extended` and `welf` are services. Each service has its “2dlf”-convertor. We ship e.g. `sendmail2dlf` and `w3c_extended2dlf`.

Slide 9

services, superservices and DLF's

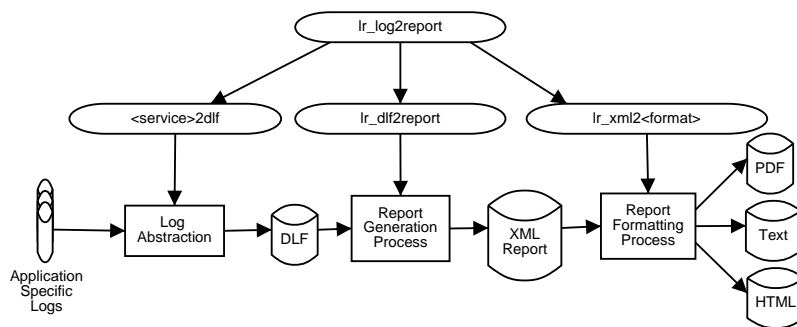
DLF "distilled log format" space separated, line oriented, fixed fields

service raw log file format

superservice a class of services, sharing same DLF and report

Slide 10

Lire's Architecture



A superservice is a class of applications which share the same DLF format, and which will give the same reports.

3.2 Receiving the log file

Slide 11

Running Lire

One can run Lire:

- as online responder
- as client
- from cron
- from command line

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 log files from other hosts, and sends generated reports back by email. The log files can be received via email or via HTTP file upload. In a client setup, the system sends log files by email to another Lire system, and receives reports back. Optionally, the logs can be anonymized before being sent. A cron driven setup reads and processes log files after they're rotated, on the local host. A userfriendly script (`lr_config`) is supplied which sets up the cronjob to your taste. In a command line driven system, users run the Lire scripts on an ad-hoc basis. One can use e.g. the `lr_log2report` script.

3.3 Converting to DLF

By invoking the right DLF convertor (e.g. `sendmail2dlf`, `cups_pagelog2dlf` or `squid-2dlf`), the log file is converted to the DLF format, suited for the superservice involved. The DLF convertor coincides with a service.

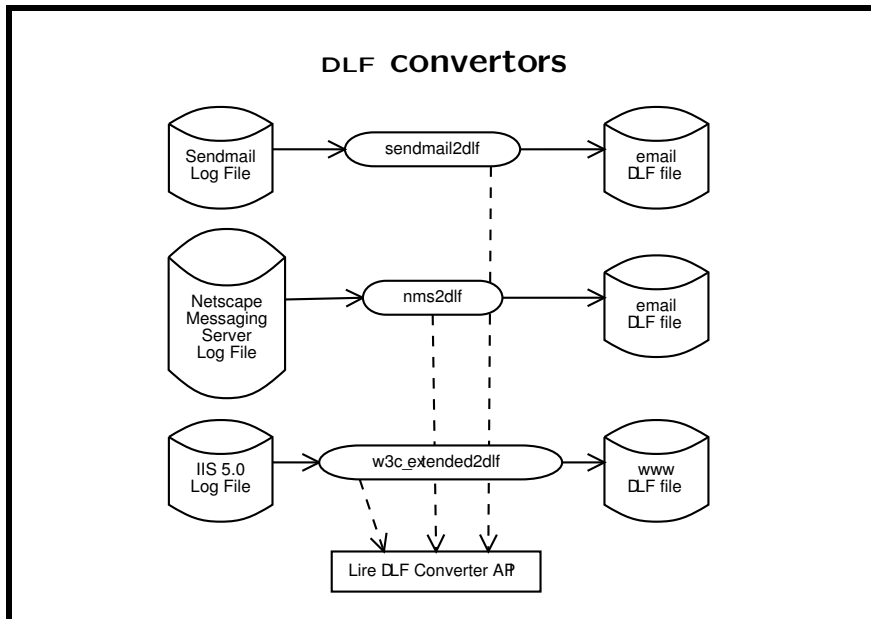
The DLF format for a superservice is defined in a Lire defined XML format. E.g., for the email superservice, this features:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lire:dlf-schema PUBLIC
  "-//LogReport.ORG//DTD Lire DLF Schema Markup Language V1.0//EN"
  "http://www.logreport.org/LDSML/1.0/ldsml.dtd">
<lire:dlf-schema superservice="email" timestamp="time"
  xmlns:lire="http://www.logreport.org/LDSML/">

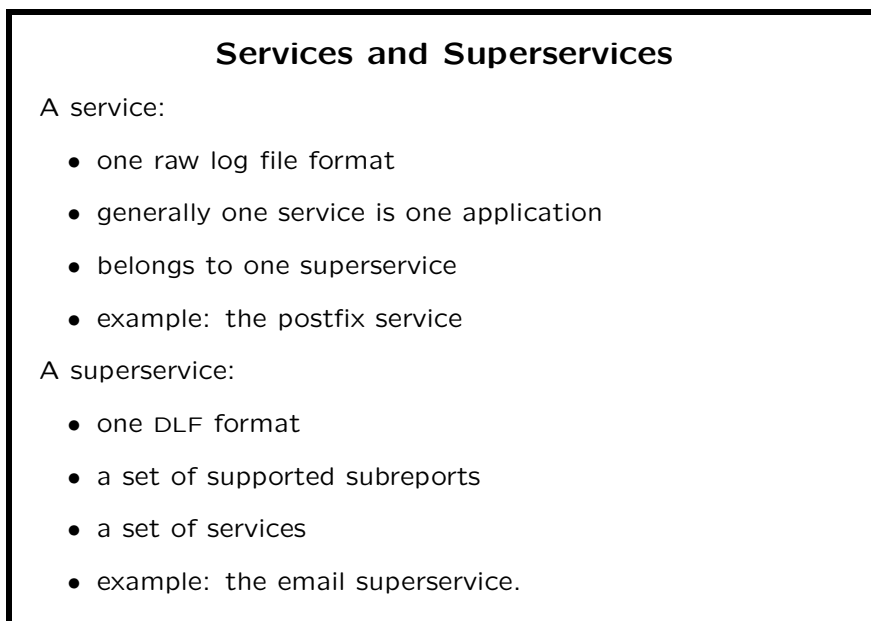
<!-- snip -->

<lire:field name="time"          type="timestamp"    default="0"/>
<lire:field name="logrelay"     type="string"      default="-"/>
```

Slide 12



Slide 13



```

<lire:field name="queueid"      type="string"      default="-"/>
<lire:field name="msgid"       type="string"      default="-"/>
<lire:field name="from_user"    type="string"      default="-"/>
<lire:field name="from_domain" type="hostname"    default="-"/>
<lire:field name="from_relay_host" type="hostname" default="-"/>
<lire:field name="from_relay_ip" type="ip"          default="-"/>
<lire:field name="size"        type="bytes"       default="0"/>
<lire:field name="delay"       type="duration"    default="0"/>

<!-- snip -->

</lire:dlf-schema>

```

Slide 14

email DLF format

```

[...]
<lire:field name="time"          type="timestamp" default="0"/>
<lire:field name="queueid"       type="string"     default="-"/>
<lire:field name="from_user"     type="string"     default="-"/>
<lire:field name="from_domain"  type="hostname"  default="-"/>
[...]

```

Please note that Lire defines its own datatypes. These are used later in the report generating mechanisms.

3.4 Generating an XML Report

A Lire report consists of several subreports, which can be displayed in graphical form, or as a table. A lot of subreports (196, as of September 2002) come with Lire predefined, but of course one can define ones own reports. A report definition is written in the Lire Report Specification Markup Language; it looks like e.g.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lire:report-spec PUBLIC
  "-//LogReport.ORG//DTD Lire Report Specification Markup Language V1.0//EN"
  "http://www.logreport.org/LRSML/1.0/lrsml.dtd">
<lire:report-spec xmlns:lire="http://www.logreport.org/LRSML/"
  superservice="email" id="top-volume-to-domain" charttype="bars">

  <lire:title>Largest Volume Sent To Domain Email Report</lire:title>
  <lire:description>
    <para>This report lists the domains to which the
      largest volume of mail was sent.</para>

```

```

</lire:description>

<lire:param-spec>
  <lire:param name="domain_to_show" type="int" default="10">
    <!-- snip -->
  </lire:param>
</lire:param-spec>

<lire:display-spec>
  <lire:title>Largest Volume Sent To Domain, Top $domain_to_show</lire:title>
  <lire:description>
    <para>Volume is in bytes</para>
  </lire:description>
</lire:display-spec>

<lire:filter-spec>
  <lire:eq arg1="$stat" arg2="sent"/>
</lire:filter-spec>

<lire:report-calc-spec>
  <lire:group sort="-mail_volume" limit="$domain_to_show">
    <lire:field name="to_domain"/>
    <lire:sum name="mail_volume" field="size"/>
  </lire:group>
</lire:report-calc-spec>

</lire:report-spec>

```

Slide 15

Report Specification

[...]

```

<lire:report-calc-spec>
  <lire:group sort="-mail_volume" limit="$domain_to_show">
    <lire:field name="to_domain"/>
    <lire:sum name="mail_volume" field="size"/>
  </lire:group>
</lire:report-calc-spec>

```

[...]

We reuse the `stat`, `to_domain`, and `size` fields from the email DLF specification. The `mail_volume` field is internal to this report; it's used only within this report calculation. Operators like `lire:group`, `lire:timegroup`, `lire:rangegroup`, `lire:timeslot`, `lire:-field`, `lire:sum`, `lire:avg`, `lire:min`, `lire:max`, `lire:count` can be used in the `report-calc-spec`.

The `domain_to_show` variable is offered as a hook for user configuration. Users can set this variable in a report configuration file.

Per superservice, there is one configuration file, for all subreports. Such a file looks like e.g.

```
# Report configuration for the email superservice

deliveries-by-period          period=1d
volume-by-period             period=1d
top-volume-to-domain         domain_to_show=10
top-to-email-by-domain       domain_to_show=30 user_to_show=5
deliveries-by-size           size=1k
deliveries-by-delay          delay-size=1s
tracked-recipients           tracked_email_re="root@example\.com"
[...]
```

The configuration file defines which subreports we want to show up in our report, and defines the settings of the variables for these subreports. For the proxy superservice, the report configuration file looks like

Slide 16

Report configuration file

```
# Report configuration for the proxy superservice

=section General
requests-summary

=section Denied Sites Reports
|select-cache_result result=TCP_DENIED
top-destinations          dsts_to_show=50
top-users-by-destinations users_to_show=8 dsts_to_show=50
[...]
```

Note the | line: this defines a filter, shared among the reports below.

The `lr_config` script, which comes with Lire, gives a userfriendly interface to set these variables.

3.5 Typesetting and publishing the Report

Graph generation is done by either the `GD::Graph` perl module or `ploticus`, a GPL-ed graphical data display engine, available from SourceForge. We show some more examples of graphs, from the `www` and `dns` superservices. Similar graphs are generated out-of-the box for the six other superservices.

3.6 Implementation

For small and medium-size logs Lire is fast. Although some heavyweight external programs are used to process XML files (jade or the fo stylesheet with PassiveTex, for typesetting PDF and RTF), performance is above expectations. When very high speed is needed, plain ascii reports can get produced. Due to its modular design, it's fairly easy to reimplement performance bottlenecks in e.g. C, to fulfill extreme performance demands.

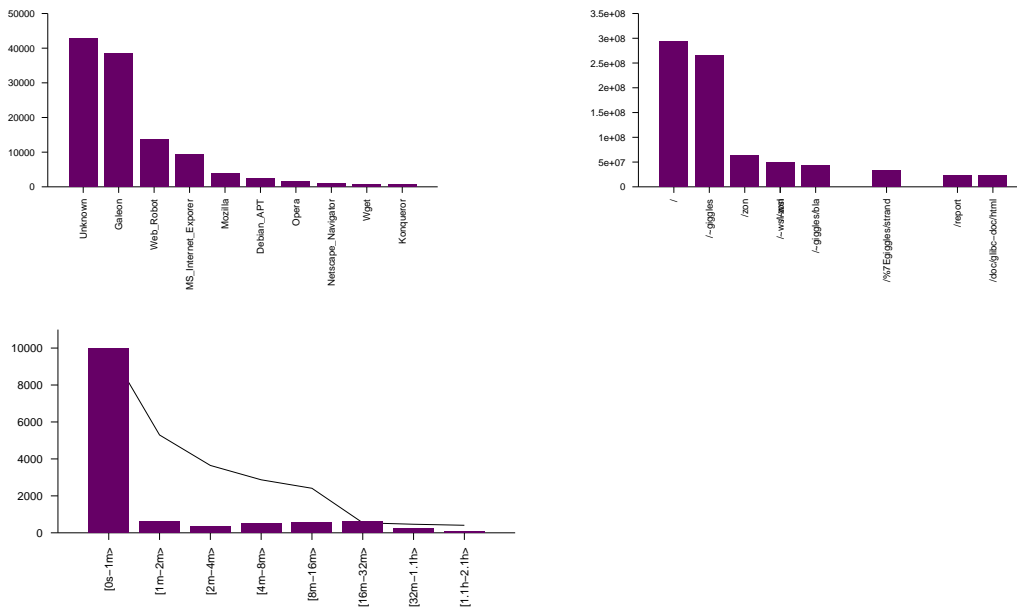


Figure 2: Reports from the www superservice: requests by browser, size by directory, user session visit times

We do have a knob (`LR_MAX_MEMORY`) to tweak the amount of memory one is willing to dedicate to Lire. This enables one to exchange disk i/o for memoryaccess.

4 Lire's future

4.1 Releases

On August 19, 2002, Lire 1.1 was released, which - again - has support for more log files.

4.2 Roadmap

We have an ambitious roadmap.

better chart generation tool In Lire 1.2, we will offer support for ploticus, next to GD::Graph. Ploticus is far more flexible, and allows for sexier graphics, with more user-configurable knobs. This code has already been committed to CVS now.

better reports It should be able to typeset multiple column reports, and generate column labels. The HTML layout should get improved.

configuration api Lire's framework should contain a configuration API that should be used by all of its components. The current mix of Perl and Shell configuration is suboptimal. Once this is reimplemented properly, we can think about a better configuration management tool (which can replace `lr_config`).

expand developers documentation We love code contributions! Therefore, it should be easy to get to learn the Lire internals: new features will get properly documented, for both users and developers.

merging and splitting of reports and log files We would like Lire to offer an easy interface to merge and split reports and log files. Currently, by default, one log file gets

Slide 19

Table of Contents

- Log file analysis
- Lire Overview
- Lire's Architecture
- Lire's Future
- The LogReport Project
- More information, contact, questions

Slide 20

Release, Roadmap

Lire 1.1 is the current release (we've published 23 releases since September 2000). But we have more plans, for Lire 1.2 (late october 2002) and on.

- better chart generation tool (1.2)
- better reports (1.2)
- configuration api (1.2)
- expand developers documentation (1.2)
- better support for merging
- cross-superservice reporting
- more services

converted to one report. It should be possible to combine different reports after the fact, e.g. for generating reports about longer timeframes, or to combine reports from different servers. It should be easy to generate handcrafted reports after the fact from already processed log files. We do offer preliminary support for this since Lire 1.0, but the default configuration doesn't yet support this. We need to implement a storage API to get this going: we need a datawarehouse infrastructure. Scheduled for Lire 1.3, december 2002.

cross-superservice reporting It will be possible to combine e.g. POP server's reports with firewall reports. This way, one can e.g. track IP's which failed to authenticate in the POP log, and see whether they triggered some firewall rules too. This feature is scheduled for Lire 2.0, late february 2003.

even more services We plan to add some more firewall convertors. Furthermore, LDAP log file support should get added. Since this task is relatively easy to do, we'd prefer to have someone not in the LogReport Developers team doing this. (The team is focusing mainly on the Lire core code.)

5 The LogReport Project

Slide 21

LogReport people

LogReport developers

- Joost van Baal
- Wessel Dankers
- Josh Koenig
- Francis Lacoste

LogReport board

- Teus Hagen (chairman)
- Wytze van der Raay (treasurer)
- Jakob Schripsema (secretary)

5.1 People working for LogReport

Four people are working for LogReport:

- Joost van Baal
- Wessel Dankers
- Josh Koenig
- Francis Lacoste

Furthermore, Joost Bekkers, Arnaud Gaillard, Edwin Groothuis and Egon Willighagen regularly contribute code to the project.

These people live and work from The Netherlands, Canada, Australia, Switzerland and the USA, all part time. Communication is done using IRC (`#logreport` on `freenode.net`) and email.

Next to a website, the project offers two mailinglists: `development@logreport.org` and `questions@logreport.org`. Both Lire and our website are maintained via CVS, hosted on SourceForge. We have our own server which hosts our website as well as the lists.

Furthermore this server hosts the LogReport Online Responder. One can send (compressed) logfiles in email messages to dedicated addresses, like e.g. `log@qmail.logreport.org`, `log@bind9.logreport.org`, and get a report back as a response. Optionally, one can anonymize the log before submitting it, using a simple script which comes with Lire.

5.2 The LogReport Foundation

Stichting LogReport Foundation is a non-profit organization; it got a legal status as a foundation, and funding by the NLnet Foundation (<http://www.nlnet.nl/>), in August 2000.

- Teus Hagen (chairman)
- Wytze van der Raay (treasurer)
- Jakob Schripsema (secretary)

5.3 How to help

Slide 22

How to help

- Use our Online Responder
- Sent (anonymized) log files
- Download Lire, and use it
- Give feedback on our mailinglists: feature requests, bug reports, help other people
- Even better: send patches and add support for other services
- Promote Lire: via webpages and mailinglists
- Fund us; buy commercial support

We need log files to test our code, and to be able to add support for more services. We especially lack log files from expensive commercial services, like the WELF and Microsoft ones. (We don't run these ourselves...)

If you use the Debian package, you can use the Debian Bug Tracking System to report bugs.

If you intend to write code, be sure to use our current CVS, of course. Our CVS, hosted on SourceForge, is readable for anyone. Code contributions of non-trivial size are accepted if – of course – they meet our quality standards, and they're offered under a GPL compatible license, like the GPL itself, or e.g. the modified BSD license. People who are willing to contribute to the Lire project during a longer time, can get write access to our CVS tree.

(Of course, since Lire is free software, you can keep your modifications private and secret too. Just be sure not to distribute them, in such a case.) Contact us for more information.

Promote Lire: link to us from your webpage, suggest using Lire on mailinglists. Join the Lire community: help other users on our lists.

Fund us: Funding from Stichting NLnet which currently enables us to spend a lot of time on Lire will run out in the future. Financial contributions to the LogReport foundation are tax-deductable under Dutch law, because LogReport is recognized as a charitable goal. Hire a LogReporter: the LogReport team is available for commercial consultancy about log file and Lire issues, see the LogReport.com website at <http://www.logreport.com/>. Contact us if you're interested.

6 More information, contact info

Slide 23

More information, contact info

website <http://www.logreport.org/>

mailing lists (archived) questions@logreport.org,
development@logreport.org

irc #logreport on freenode.net

announcements announcement@logreport.org

Questions?

6.1 More information

More information is on our website on <http://www.logreport.org/>. We have several mailing lists, which are archived. A lot of documentation and manpages come with Lire.

We sent newsitems via our announcement@logreport.org list. Subscribe to it if you wanna be kept informed.

6.2 Contact

Contact us via our lists questions@logreport.org and development@logreport.org (see our website for subscription info), or privately on logreport@logreport.org. To have an informal chat with the LogReport developers, join the #logreport IRC channel on the Open Projects Network.