LILA - **L**ive **I**ptables **L**og **A**nalyzer

Documentation

Contents

1 Feature overview

- Convert unreadable textlog garbage into a nice colored and easy to read output.
- Live monitor packets and don't show duplicate ones, to avoid output spam. (This increases readability a lot).
- Instantly see if a program wants to "phone home" for example. (Or has already sent packets, depending on your firewall rules).
- Show the hostname of the IP, so you can instantly see that a packet belongs to www.google.com for example.
- Separate config file with many options, so there is no need to dive into LILA's source code to customize it.
- Customizable duplicate detection time interval, which LILA uses to consider a packet a dupe. Time interval can be specified using days, hours, minutes and seconds. Entering 1d5h2m1s is obviously more convenient than to specify 104521 seconds.
- Set up individual output color rules and what is being displayed and what not. The datetime format can be changed, too.
- Show newly added (unviewed) entries in the logs since the last start of LILA or show the last n entries. (Duplicate filter still active by default).
- Highly portable. Syslog database can be on computer 1, LILA's database on computer 2 and LILA itself can be run on computer 3 if you like.
- Automatically resolve hostnames for IPs and add them to the database for instant lookups for the next time, the same IP is detected. LILA uses two different techniques to accomplish reverse DNS. (More in section DESCRIPTION below).
- You can freely choose between up to four available resolvers. All combinations are possible, though dig, host and system are esentially the same. Use of pdnsd is highly (!) recommended in order to get the "best" hostname, i.e. the one, who was the actual answer of the initial DNS request when the packet was logged.
- LILA additionally keeps a local DNS cache to further reduce lookups in the DNS table.
- DNS caching time can be changed in the config file using hours, days, minutes, seconds as format.
- LILA now keeps track of hostname (DNS) changes. If you're analyzing older logs (i.e. DNS records have changed) LILA automatically chooses the right record (the one which was valid at the time the packet was logged) from its database. It's no problem to see whether and when a certain IP has changed the hostname (resp. vice versa).
- Support for blacklisting entries, which won't show up.
- Simply search your log files by keyword (ip, hostname, port etc.).
- View and delete tables created by LILA or syslog. (current, dns, staticfile tables).
- SSH support: It's possible to run LILA on your (fast) main computer, while pdnsd resolving is performed via SSH on the firewall computer. The MySQL database is stored on the local (faster) computer. This results in a speedup, especially when your main computer is faster than your firewall, what is usually the case.
- Advanced search modes makes it possible to specify complex search queries. [not yet backported from 0.8 and later].
- View detailled statistics of your logs. (TOP 10 or TOP x in each category) [not yet backported from 0.7.5 and later].
- Archive / backup functionality. It's possible to have a steadily growing backup table which contains your everyday logs. LILA automatically detects new entries. It's no problem when your current logfile is completely (for example after a reboot, if your logs are saved in RAM) or partially deleted. LILA detects such "non-continuous" logfiles, adds only new entries and remembers the new (changed) line position, so the next time you start a backup the logile is again considered continuous. [not yet backported from 0.6 and later].

- Automatically monitor one or more specific IPs and save the packets to a separate table to analyze them later. [not yet backported from 0.8.5 and later].
- Analyze and save static logfiles. [not yet backported from 0.4 and later].
- Please type lila `--help` or `lila -?` to get a brief overview of the features and or read the changelog.
- more...

2 Description

LILA is a command line tool written in Python for (live) analyzing iptables firewall logs. It uses a MySQL database and adds one or more reverse IP entries to a DNS table.

It uses two techniques to resolve IPs to hostnames:

- By simply making a PTR query to a public DNS server using `dig`, `system` or `host`
- By querying the cache of the own local DNS server `pdnsd` (this usually corresponds to the initial DNS request of the browser).

As DNS takes some time, especially with large (1000-10000+ entries) log files, the results are saved into a separate MySQL table, so next time resolving will be A LOT faster. Additionally LILA keeps a local DNS session cache.

At the moment LILA just resolves destination IPs as I am using it on a bridged firewall behind a router, that blocks incoming packets. Currently its main purpose is to monitor outgoing packets by live monitoring the system log to see what's currently being blocked or allowed. It produces a clear colored output which only contains the most important data and is easy to read. By standard it doesn't list duplicate ("dupe") entries, if the IP occured in the last 60 seconds. The time interval can be changed by command line or by config file, like several other options, too. The reason is to prevent "spam output" of dozens of identical IPs (can be deactivated). Of course the database does always contain every single logged packet, for later analysis. They're just not printed to screen.

If you missed starting LILA and still want to know which packets just got blocked you can simply tell it by command line to list the last n logs or tell it to list every log since the last start. After that it continues as usual with live monitoring.

Additionally it's possible to list log entries from static log files. For example if one has a cronjob which splits (by date, size etc.) the iptables logs into several smaller files/tables. LILA creates an individual table for each file/table (using an MD5 hash), so next time the logs are being analyzed by LILA, it doesn't need to do the same actions again (creating the database and DNS resolving). This way it's possible to review log files again without being forced to wait. Again this results in a BIG speedup on large logfiles.

For your convenience and to speed up LILA you can run it on your main computer, while the logs reside on a different machine (server / firewall whatever).

Its main purpose is to see "what's currently happening on your network". You might be surprised that programs send packets to servers you've never heard of. For example firefox continuously sends information to google's safebrowsing servers by default. If you have an external firewall you can also test your software firewall or when using Microsoft Windows see how many programs want to "phone home".

If there are any remaining questions, comments or anything else please feel free to contact me at any time. It's always nice to get feedback :-)

Happy monitoring!

3 LILA in action (with screenshots)

LILA in live log mode. You can see several accepted outgoing UDP packets with the destination port 123 (ntp). Obviously the NTP daemon seems to prefer time servers located in the Netherlands. Additionally an ICMP echo request (icmp8) has been sent to the (whitelisted) server www.gentoo.org, which answers with an echo reply (icmp0) message. An attempt to visit http://www.ofdb.de and http://www.malware.com has been blocked. You can see from LILA's reverse DNS functionality, that those sites are hosted at netbuild.net resp. megawebservers.com. By the way: The "nice" hostnames come from pdnsd!

```
jaf@argentum ~ $ lila
============================================================
   LILA - Live Iptables Log Analyzer - version 1.0
============================================================

 * Existing table found (239 packets).

 * Adding 5 new syslog entries... Done!

 * Hide duplicate packets. Dupe time interval set to 3600 seconds.

 * Starting live log monitoring... Done! (Press CTRL + C to exit LILA)

03-15 15:12:28 ACCEPTntp 192.168.2.5:123     ---udp---> 83.98.201.134:123    ntp1.mediamatic.nl
03-15 15:12:37 Whitelist 192.168.2.5         --icmp8--> 89.16.167.134        www.gentoo.org
03-15 15:12:37 ACCEPT    89.16.167.134       --icmp0--> 192.168.2.5          argentum
03-15 15:13:34 ACCEPTntp 192.168.2.5:123     ---udp---> 109.72.80.61:123     virtueledoos.nl
03-15 15:13:42 ACCEPTntp 192.168.2.5:123     ---udp---> 91.148.192.49:123    sip.dicode.nl
03-15 15:13:49 DROP      192.168.2.5:58631   ---tcp---> 87.237.120.168:80    nb2520.virtualhosts.netbuild.net, www.ofdb.de
03-15 15:14:39 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:55 DROP      192.168.2.5:46737   ---tcp---> 216.251.32.98:80     hosting.megawebservers.com, www.malware.com
_
```

If you're interested in the last 20 logs you can enter `lila -q -n 20`. Option `-q` tells LILA not to enter the live monitoring mode after analyzing the last 20 packets. You can see that only 5 of 20 packets are displayed, the other 15 are hidden because they are duplicates:

```
jaf@argentum ~ $ lila -q -n 20
============================================================
   LILA - Live Iptables Log Analyzer - version 1.0
============================================================

 * Existing table found (284 packets).

 * Hide duplicate packets. Dupe time interval set to 3600 seconds.

 * Analyzing last 20 packets...

03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:55 DROP      192.168.2.5:46737   ---tcp---> 216.251.32.98:80     hosting.megawebservers.com, www.malware.com
03-15 15:15:44 ACCEPTntp 192.168.2.5:123     ---udp---> 109.72.80.61:123     virtueledoos.nl
03-15 15:15:51 ACCEPTntp 192.168.2.5:123     ---udp---> 91.148.192.49:123    sip.dicode.nl
03-15 15:16:46 ACCEPTntp 192.168.2.5:123     ---udp---> 83.98.201.134:123    ntp1.mediamatic.nl

 * 5 of 20 packets shown (15 hidden).

 * Exiting LILA... Goodbye!

jaf@argentum ~ $ _
```

Of course you can tell LILA to show all packets. Just specify command line option `-n` or `--showdupes`:

```
jaf@argentum ~ $ lila -q -n 20 -d
============================================================
   LILA - Live Iptables Log Analyzer - version 1.0
============================================================

 * Existing table found (284 packets).

 * Show all packets, including dupes.

 * Analyzing last 20 packets...

03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:40 WALLadd   192.168.2.5:50079   ---tcp---> 149.20.20.133:80     pub1.kernel.org
03-15 15:14:55 DROP      192.168.2.5:46737   ---tcp---> 216.251.32.98:80     hosting.megawebservers.com, www.malware.com
03-15 15:15:44 ACCEPTntp 192.168.2.5:123     ---udp---> 109.72.80.61:123     virtueledoos.nl
03-15 15:15:51 ACCEPTntp 192.168.2.5:123     ---udp---> 91.148.192.49:123    sip.dicode.nl
03-15 15:16:46 ACCEPTntp 192.168.2.5:123     ---udp---> 83.98.201.134:123    ntp1.mediamatic.nl
03-15 15:17:53 ACCEPTntp 192.168.2.5:123     ---udp---> 109.72.80.61:123     virtueledoos.nl
03-15 15:17:58 ACCEPTntp 192.168.2.5:123     ---udp---> 91.148.192.49:123    sip.dicode.nl

 * 20 of 20 packets shown (0 hidden).

 * Exiting LILA... Goodbye!

jaf@argentum ~ $ _
```

If you forgot to start LILA and want to know which packets have been sent, use the command line option `-c` or `--listnew`. This tells LILA to list all packets since its last start. Here you can see that I visited the gentoo wiki and that firefox wanted to communicate with google's safebrowsing servers. You can also see a small drawback of LILA 1.0: The line break could look better. This will be addressed in later versions.

```
jaf@argentum ~ $ lila -c
=================================================
   LILA - Live Iptables Log Analyzer - version 1.0
=================================================

 * Existing table found (750 packets).

 * Adding 16 new syslog entries... Done!

 * Hide duplicate packets. Dupe time interval set to 3600 seconds.

 * Analyzing packets since the last start...

03-15 16:20:58 ACCEPT    192.168.2.5:36194    ---tcp---> 207.98.216.138:80    static-207-98-216-138.knology.net, gentoo-wiki.com
03-15 16:21:05 DROP      192.168.2.5:36457    ---tcp---> 209.85.148.139:80    fra07s07-in-f139.1e100.net, safebrowsing.cache.l.g
oogle.com
03-15 16:21:06 DROP      192.168.2.5:60366    ---tcp---> 209.85.148.100:80    fra07s07-in-f100.1e100.net, safebrowsing.cache.l.g
oogle.com

 * 3 of 16 packets shown (13 hidden).

 * Starting live log monitoring... Done! (Press CTRL + C to exit LILA)
```

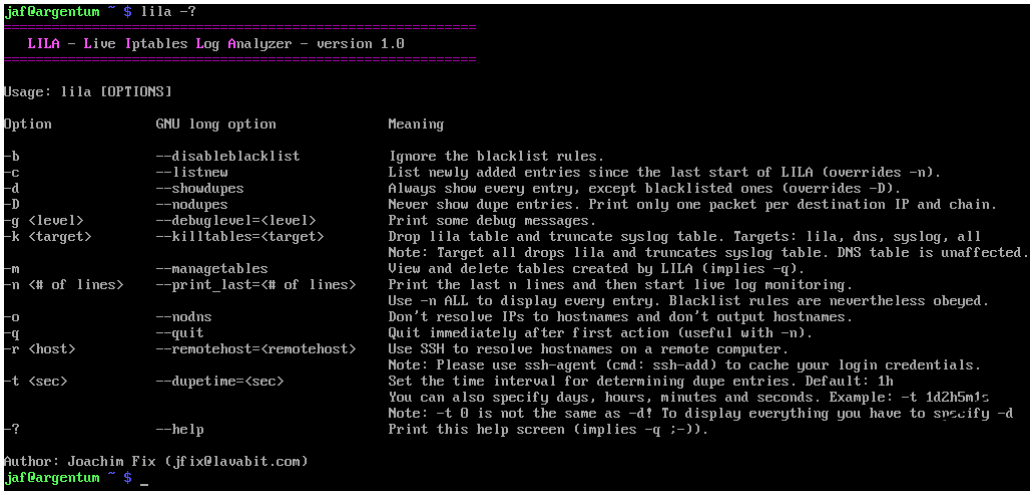
For the moment, we can just change the way how LILA outputs data. Let's modify the config file and disable the system resolver to get rid of the ugly PTR record. Furthermore we can set `sourceip_format` to `hostname` instead of `ip`. Additionally we don't want to show the destination port and as we have gained some space now, why not change the datetime format to also show the year? You might prefer the destination hostnames to be printed in white instead of yellow, so we also modify the output color rules. Because 16 packets have been added, we start LILA again with `lila -n 16 -q` and get the following output:

```
jaf@argentum ~ $ lila -n 16 -q
=================================================
   LILA - Live Iptables Log Analyzer - version 1.0
=================================================

 * Existing table found (766 packets).

 * Hide duplicate packets. Dupe time interval set to 3600 seconds.

 * Analyzing last 16 packets...

2011-03-15 16:21:57 ACCEPT    argentum:36197 ---tcp---> 207.98.216.138       gentoo-wiki.com
2011-03-15 16:22:00 DROP      argentum:60368 ---tcp---> 209.85.148.100       safebrowsing.cache.l.google.com
2011-03-15 16:22:00 DROP      argentum:40368 ---tcp---> 209.85.148.138       safebrowsing.cache.l.google.com

 * 3 of 16 packets shown (13 hidden).

 * Exiting LILA... Goodbye!

jaf@argentum ~ $
```

...and here are the same 16 packets in their original state (raw netfilter log, reverse order):

```
jaf@argentum ~ $ mysql -u syslog -psyslogpw -s -e "SELECT msg FROM syslog.netfilter_logs ORDER BY id DESC LIMIT 16"
msg
[31964.498777] iptables: DROP IN= OUT=eth0 SRC=192.168.2.5 DST=209.85.148.138 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=57006 DF PROTO
=TCP SPT=40368 DPT=80 WINDOW=4380 RES=0x00 SYN URGP=0 OPT (020405B40402080A0078CA5C0000000001030306) UID=0 GID=0
[31963.705856] iptables: DROP IN= OUT=eth0 SRC=192.168.2.5 DST=209.85.148.100 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=6606 DF PROTO=
TCP SPT=60368 DPT=80 WINDOW=4380 RES=0x00 SYN URGP=0 OPT (020405B40402080A0078C9960000000001030306) UID=0 GID=0
[31961.699007] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28111 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=690 RES=0x00 ACK URGP=0 OPT (0101080A0078C7A028E80E8F)
[31961.563798] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28110 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=690 RES=0x00 ACK FIN URGP=0 OPT (0101080A0078C77E28E80E6D) UID=0 GID=0
[31961.562055] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28109 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=645 RES=0x00 ACK URGP=0 OPT (0101080A0078C77E28E80E6D) UID=0 GID=0
[31961.435831] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28108 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=555 RES=0x00 ACK URGP=0 OPT (0101080A0078C75E28E80E4D) UID=0 GID=0
[31961.434016] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28107 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=465 RES=0x00 ACK URGP=0 OPT (0101080A0078C75E28E80E4D) UID=0 GID=0
[31961.430006] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28106 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=375 RES=0x00 ACK URGP=0 OPT (0101080A0078C75D28E80E4B) UID=0 GID=0
[31961.301431] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28105 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=285 RES=0x00 ACK URGP=0 OPT (0101080A0078C73D28E80E2A) UID=0 GID=0
[31961.296361] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28104 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=195 RES=0x00 ACK URGP=0 OPT (0101080A0078C73C28E80E2A) UID=0 GID=0
[31961.161939] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=182 TOS=0x00 PREC=0x00 TTL=64 ID=28103 DF PR
OTO=TCP SPT=36197 DPT=80 WINDOW=105 RES=0x00 ACK PSH URGP=0 OPT (0101080A0078C71A28E80E0B) UID=0 GID=0
[31961.027394] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=168 TOS=0x00 PREC=0x00 TTL=64 ID=28102 DF PR
OTO=TCP SPT=36197 DPT=80 WINDOW=87 RES=0x00 ACK PSH URGP=0 OPT (0101080A0078C6F828E80DE0) UID=0 GID=0
[31961.001381] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28101 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=87 RES=0x00 ACK URGP=0 OPT (0101080A0078C6F228E80DE0) UID=0 GID=0
[31960.863947] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=165 TOS=0x00 PREC=0x00 TTL=64 ID=28100 DF PR
OTO=TCP SPT=36197 DPT=80 WINDOW=69 RES=0x00 ACK PSH URGP=0 OPT (0101080A0078C6CF28E80DBE) UID=0 GID=0
[31960.863654] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=28099 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=69 RES=0x00 ACK URGP=0 OPT (0101080A0078C6CF28E80DBE) UID=0 GID=0
[31960.729558] iptables: ACCEPT IN= OUT=eth0 SRC=192.168.2.5 DST=207.98.216.138 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=28098 DF PRO
TO=TCP SPT=36197 DPT=80 WINDOW=4380 RES=0x00 SYN URGP=0 OPT (020405B40402080A0078C6AE0000000001030306) UID=0 GID=0
jaf@argentum ~ $ _
```

Last but not least LILA's help screen:



4 Installation

Note: Generally if you encounter any problems feel free to ask me. I'll try to help you and update the documentation accordingly.

4.1 Contents

Downloadlink: `https://sourceforge.net/projects/lila/files/lila-1.0.tar.gz/download`
Documentation: `https://sourceforge.net/projects/lila/files/lila.pdf/download`

MD5 hashes of lila-1.0:

lila `66518aee1bee40bfc03c0b45df300a21`
lila.cfg `5bd566627c6151598974346de916565a`

4.2 Prerequisites

- Urge to know what's currently happening (or has happened) on your network.
- Syslog-ng 3.0 or higher (tested with 3.1.4)
- iptables (kernel must support netfilter/iptables)
- MySQL (tested with 5.1.51)
- Python (tested with 2.6.6)
- Python-MySQL

Optional (highly recommended!):

- pdnsd (local caching DNS server, tested with 1.2.8)
- sudo (to access pdnsd's cache as normal user)

For gentoo users:

Gentoo packages: dev-lang/python, dev-python/mysql-python, net-firewall/iptables, app-admin/syslog-ng, dev-db/mysql, net-dns/pdnsd, app-admin/sudo

One liner: emerge -av python mysql-python iptables syslog-ng mysql pdnsd sudo

4.3 Setup of the MySQL databases

- The following instructions assume that the syslog-ng and LILA databases are both on localhost, if this is not the case I bet you already know what to do instead.
- Note: Most parts of the following commands are taken from the Gentoo wiki:
  `http://en.gentoo-wiki.com/wiki/Syslog-ng_directly_to_MySQL` (Thank you!)
- Install and start the MySQL server. To initially setup MySQL, enter `mysql_secure_installation` at the command line.
- Connect to the server using the command `mysql -u root -p` and enter the password you've set in the step before.
- Create a MySQL database and table, where syslog-ng will log all raw netfilter messages:
- Here is an example how to create them. LILA itself only uses the fields `id`, `datetime` and `msg`, so feel free to customize (resp. shorten) the following commands:
  Note: You should change the `SET PASSWORD` commands to use reasonable passwords.

```
CREATE DATABASE ‘syslog‘ DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
CREATE TABLE IF NOT EXISTS ‘syslog‘.‘netfilter_logs‘(
‘id‘ BIGINT(20) unsigned NOT NULL auto_increment,
‘host‘ VARCHAR(128) collate utf8_unicode_ci DEFAULT NULL,
‘facility‘ VARCHAR(10) collate utf8_unicode_ci DEFAULT NULL,
‘priority‘ VARCHAR(10) collate utf8_unicode_ci DEFAULT NULL,
‘level‘ VARCHAR(10) collate utf8_unicode_ci DEFAULT NULL,
‘tag‘ VARCHAR(10) collate utf8_unicode_ci DEFAULT NULL,
‘datetime‘ datetime DEFAULT NULL,
‘program‘ VARCHAR(15) collate utf8_unicode_ci DEFAULT NULL,
‘msg‘ TEXT collate utf8_unicode_ci,
‘seq‘ BIGINT(20) UNSIGNED NOT NULL DEFAULT ’0’,
‘counter‘ INT(11) NOT NULL DEFAULT ’1’,
‘fo‘ datetime DEFAULT NULL, ‘lo‘ datetime DEFAULT NULL,
PRIMARY KEY (‘id‘), KEY ‘datetime‘ (‘datetime‘), KEY ‘sequence‘ (‘seq‘),
KEY ‘priority‘ (‘priority‘), KEY ‘facility‘ (‘facility‘),
KEY ‘program‘ (‘program‘), KEY ‘host‘ (‘host‘)  )
ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
CREATE DATABASE ‘lila‘ DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
GRANT SELECT , INSERT , UPDATE , DELETE , CREATE , DROP , INDEX , ALTER ON
‘syslog‘ . * TO ’syslog’@’localhost’;
GRANT SELECT , INSERT , UPDATE , DELETE , CREATE , DROP , INDEX , ALTER ON
‘lila‘ . * TO ’lila’@’localhost’;
SET PASSWORD FOR ’syslog’@’localhost’ = PASSWORD( ’syslogpw’ );
SET PASSWORD FOR ’lila’@’localhost’ = PASSWORD( ’lilapw’ );
```

- Enter `EXIT;` to leave the MySQL command line.

4.4 Setup of syslog-ng

- Install syslog-ng 3 with MySQL support and add it to the default runlevel so it will be started at boot time.
- Edit syslog-ng.conf (gentoo: /etc/syslog-ng/syslog-ng.conf) and add the following:

```
destination d_netfilter_mysql {
        sql(type(mysql)
        host("localhost") username("syslog") password("syslogpw")
        database("syslog")
        table("netfilter_logs")
        columns("host", "facility", "priority", "level", "tag",
                "datetime", "program", "msg", "seq")
        values("$HOST_FROM", "$FACILITY", "$PRIORITY", "$LEVEL", "$TAG",
               "$YEAR-$MONTH-$DAY $HOUR:$MIN:$SEC", "$PROGRAM", "$MSG", "$SEQNUM")
        indexes("host", "facility", "priority", "datetime", "program", "seq"));
};
filter f_netfilter { match("iptables:" value("MESSAGE")); };
log { source(src); filter(f_netfilter); destination(d_netfilter_mysql); };
```

  Note: If you don't want the netfilter logs to appear on tty12 (syslog-ng default) change the line containing the log directive to:

```
log { source(src); filter(f_netfilter); destination(d_netfilter_mysql); flags(final); };
```

  and add it before every other log directive.

- Make sure that syslog-ng is running (e.g. /etc/init.d/syslog-ng restart)

4.5 Setup of pdnsd

- Note 1: Setup of pdnsd is optional, but it is *highly* recommended in order to get the "best" DNS records. If you decide to skip this step, disable pdnsd in LILA's config file or run LILA with the `--nodns` option to completely disable DNS functionality.
- Note 2: To avoid problems you might want to disable IPv6 support. It can lead to error messages about ports or permissions in the syslog. When encountering problems consult the pdnsd documentation or drop me an email.
- The following instructions describe how to setup pdnsd for personal use with opendns servers. If you have other needs, the setup of pdnsd's config file is very easy, because in most cases you only have to comment out the part that you want. Besides, the config file contains various comments and a very good documentation/manpage is available, too.
- It's enough to comment out the opendns part and to comment the rest, but for the sake of completeness here is my config file, which is more or less identical to the sample config file with some minor changes and unnecessary parts removed. You can just copy&paste it if you like.

  /etc/pdnsd/pdnsd.conf:

```
global {
perm_cache=2048;
cache_dir="/var/cache/pdnsd";
# pid_file="/var/cache/pdnsd/pdnsd.pid";
run_as="pdnsd";
ctl_perms=0600;
server_ip = 127.0.0.1;  # Use eth0 here if you want to allow other
status_ctl = on;
# paranoid=on;       # This option reduces the chance of cache poisoning
 but may make pdnsd less efficient, unfortunately.
query_method=udp_tcp;
min_ttl=15m;       # Retain cached entries at least 15 minutes.
max_ttl=1w;        # One week.
timeout=10;        # Global timeout option (10 seconds).
neg_rrs_pol=auth;
neg_domain_pol=auth;
}

server {
label = "opendns";
ip = 208.67.222.222, 208.67.220.220;
reject = 208.69.32.0/24, 208.69.34.0/24, 208.67.219.0/24; #opendns search engine redirection
reject_policy = negate;
timeout = 4;
uptest = none;
preset = on;
}

source {
owner=localhost;
#serve_aliases=on;
file="/etc/hosts";
}

rr { name=localhost; reverse=on; a=127.0.0.1; owner=localhost;
     soa=localhost,root.localhost,42,86400,900,86400,86400; }
# neg { name=doubleclick.net; types=domain; }   # This will also block xxx.doubleclick.net...
# neg { name=bad.server.com; types=A,AAAA; } #  Bad server you don't want to connect to.
```

- LILA needs the privilege to run the command `pdnsd-ctl dump` so you have to add the following line to your /etc/sudoers (type visudo as root to edit the file):

  ```
  $USER ALL=(pdnsd) NOPASSWD: /usr/sbin/pdnsd-ctl dump
  ```

- Note: You have to replace `$USER` with the username used to run LILA! You might also need to change the path where pdnsd-ctl is located.

  What this does is allow LILA to run this specific command as user "pdnsd". It allows LILA only to read the DNS cache; there is no way to change DNS records or anything else and no root privileges are granted at any time!

- Run pdnsd e.g. with `/etc/init.d/pdnsd restart` (depends on your distribution).

- Test functionality with you user account by enetring `sudo -u pdnsd /usr/sbin/pdnsd-ctl dump` at the command line.

- Add `nameserver 127.0.0.1` at the beginning of /etc/resolv.conf.

- If you use DHCP you probably want to add something like:

  ```
  interface eth0
  static domain_name_servers=127.0.0.1
  ```

  to your /etc/dhcpcd.conf or like described in `http://ubuntuforums.org/showthread.php?t=331850` comment out the line

  ```
  #prepend domain-name-servers 127.0.0.1;
  ```

  in /etc/dhcp3/dhclient.conf, to ensure that your settings are still there the next time you reboot.

- Add pdnsd to the default runlevel so it will be started at boot time.

4.6 Creating iptables chains

- You have to specify log prefixes for your iptables chains. They must start with `"iptables: "`, unless otherwise specified in syslog-ng.conf and under section GENERAL in lila.cfg.

- Example: (note the whitespace at the end of the log-prefix!)

  ```
  IPTABLES="/sbin/iptables"
  $IPTABLES -N LOGACCEPT
  $IPTABLES -A LOGACCEPT -j LOG --log-prefix "iptables: ACCEPT " --log-level debug \\
    --log-ip-options --log-tcp-options --log-uid
  $IPTABLES -A LOGACCEPT -j ACCEPT
  $IPTABLES -N LOGDROP
  $IPTABLES -A LOGDROP -j LOG --log-prefix "iptables: DROP " --log-level debug \\
    --log-ip-options --log-tcp-options --log-uid
  $IPTABLES -A LOGDROP -j DROP
  # Drop and log packets with destination www.website.com (65.61.199.238)
  $IPTABLES -A OUTPUT -p all -d 65.61.199.238 -j LOGDROP
  # Accept and log packets with destination gentoo.org (89.16.167.134)
  $IPTABLES -A OUTPUT -p all -d 89.16.167.134 -j LOGACCEPT
  ```

4.7 Edit LILA's configuration file

- You have to enter your MySQL login credentials and choose the resolvers you want to use. Default is pdnsd and system.

- See the comments in lila.cfg for more details.

- Note: LILA's configuration file must be in the same directory as LILA and use the same name as the executable with the extension .cfg!

4.8 Starting LILA

- Change into the directory where lila and lila.cfg are located.
- Make sure LILA is executable (it should already be) by enetering `chmod +x lila`
- Start LILA with `./lila` or create a symlink with `ln -sf $(pwd)/lila /usr/bin/lila` to start LILA from everywhere by just entering `lila`.

5 Changelog

LILA 1.0

- Code has been completely rewritten for better performance and simplicity. Easy to understand variable names and built-in debug messages simplify code changes and debugging.
- LILA doesn't analyze text logfiles anymore, but MySQL log files, like created by syslog-ng version 3 or higher. This heavily increases the performance and allows to do analyses based on the packet's datetime information which is also logged by syslog-ng. Perhaps future versions will backport text logfile analysis like in LILA versions <1.0.
- Rarely used features and beta features have been stripped. (Maybe future versions will backport some of the old functionality.)
- Nicer and easier to understand status messages. Output design inspired by Gentoo's init.d/ebuild system (thank you).
- New way how reverse DNS resolving works and how it is configured:
  - You can freely choose between up to four available resolvers. All combinations are possible, though dig, host and system are esentially the same. Use of pdnsd is highly (!) recommended in order to get the "best" PTR record, i.e. the one, who was the actual answer of the initial DNS request when the packet was logged.
  - The resulting PTR string, which contains the duplicate-free results of all chosen resolvers now isn't saved to LILA's database by default, resulting in a big speedup.
  - DNS queries are by default only performed in the live monitoring mode. LILA caches the PTR records it finds in a DNS table to speed up lookups in future sessions.
  - LILA additionally keeps a local DNS cache to further reduce lookups in the DNS table.
  - DNS caching time can be changed in the config file.
  - LILA now keeps track of PTR record changes. If you're analyzing older logs (i.e. DNS records have changed) LILA automatically chooses the right record (which was valid at the time the packet was logged) from its database.
- Detection of the lila configfile path now uses realpath, to avoid problems with symlinks.
- Updated manual / install notes (including configuration of iptables, syslog-ng and pdnsd).

0.8.5-beta

- New feature: Automatically monitor one or more specific IPs and save the packets to a separate table to analyze them later.
- Disabled warning for MySQL 5.1 and later about log formats.

0.8.3-beta

- Fixed "December" bug.
- Added killswitch -k which automatically deletes the locally stored logfile, before LILA starts. (sudoers must be configured accordingly).
- Added commented experimental source code, that doesn't alter any functionality at the moment.
- Various minor bug fixes and code changes.

0.8-beta

- New feature: Advanced search queries. (Command line option -S).
- Added resolver3 (system) which can be used when resolver1 or resolver2 got no result (or are deactivated) or when their results are identical.
- Fixed an error when a non-continuous log file was empty.
- Arrow keys and history now work properly on input fields.
- Command line option `print_last` now has an effect, when combined with search functions.
- Some minor code and output text changes.

0.7.5-beta

- You can choose between different resolvers in the config file. Added "host" as an alternative to "dig". (Possible combinations are: pdnsd/dig, pdnsd/host, —/dig, —/host and —/—)
- It's now possible to view stats of the TOP x instead of the TOP 10. (lila config file)
- It's now possible to specify the dupe time interval, using d, h, m and s. For example -t 2h3m1s means 2 hours 3 minutes and 1 second. If none of the abbrevations is used LILA interprets the value (like before) as seconds.
- LILA can now automount your remote logdir as an SSH filesystem if you specify it in the config file.
- Added info outputs:
- When LILA starts the live logging mode.
- Number of lines that are shown / hidden when viewing old packets. (E.g. with -n ¡number¿).
- Number of packets stored in the current table.
- Added support for the IGMP protocol.
- Updated Shebang! (Magic Line) from `#!/usr/bin/python` to `#!/usr/bin/env python.`
- Fixed a DNS resolving error for the top IPs. Reordered stats table: Dest. IP is now in the first column, followed by dest. port, chain, protocol...
- Fixed an error that occurred when the `print_last` argument was greater than the number of existing logs.

0.7-beta

- Added statistics. LILA can now output a TOP 10 list of every entry in the database. You can choose the desired table (current, backup, staticfile tables) and get a nice tabular overview of what's happening on your network. Questions like which is the most common destination IP or port are now answered within one table. If the backup table is chosen, you'll also see the dates of non-continuous jumps, that occur when you delete old logs or when they are deleted because you rebooted the logging computer and the logs are stored in RAM. Command line option is -i or –stats. Try it!
- Fixed a formatting error in the manage tables interface.
- Changed the datatypes stored in the MySQL table. (Database size decreased. Speed increased?) (Reduced the maximum length of chain names to 12, not counting the IPTABLES- prefix. –logprefix "IPTABLES-NTPAcceptLOG " is an example for the maxium allowed length, any longer names are still possible but will be trimmed.
- LILA now creates the backup/archive table automatically if it doesn't yet exist.
- Fixed an issue, where creating archive tables took much longer than usual.
- Fixed an error that occured, when changing the name of the backup table in lila.cfg.
- LILA now displays the total number of backup entries when it detects a continuous logfile.

0.6.5-beta

- It's now possible to search staticfiles and the backup table. (Use -f BACKUP -s ¡keyword¿).

- Restructured LILA's output function and fixed a formatting issue for long source IPs. ==¿ Increased output speed when LILA is told to ignore dupes.
- Staticfile table names now begin with `static_` (fixed an issue, where LILA couldn't create a table, when the first digit of the file's md5 hash is a number).
- Fixed: When specifying a search keyword LILA always showed dupe entries.
- Added command line option -D to completely ignore dupes. LILA prints only one packet per destination IP and chain. (Better than specifying -t 99999999).
- Updated help screen and lila.cfg.
- Some minor code changes.

0.6-beta

- Added archive functionality. It's now possible to have a steadily growing backup table which contains your everyday logs. LILA automatically detects new entries. It's no problem when your current logfile is completely (for example after a reboot, if your logs are saved in RAM) or partially deleted. LILA detects such "non-continuous" logfiles, adds only new entries and remembers the new (changed) line position, so the next time you start a backup the logile is again considered continuous. (Command line option is -a)
- Removed .py extension.
- Several internal code changes.

0.5-beta

- You can now search your logs (and DNS entries) for a specific keyword. (destination ip, hostname etc.)
- It's now possible to set individual output color rules in the config file.
- It's now possible to set up blacklist rules (by ip, chain etc.) in the config file, to prevent LILA from outputting the concerning logs. Blacklist rules can be ignored by command line option -b.
- When analyzing static logfiles LILA did not omit dupe entries, regardless which time interval was chosen.
- LILA does no longer consider a certain IP which has changed the chain while LILA is running a dupe.
- Added text what LILA does / finds, when using a static logfile.
- Added LICENSE file.
- Some minor code changes + changed some command line letters.

0.4.4-beta

- The MySQL table ids of the current table are no longer auto incremented. They are managed by LILA itself for better consistency.
- Simplified the way how LILA waits for a change on the live logfile.

0.4.2-beta

- Fixed an issue, where LILA wrongly identified a known live logfile as a new one, thus creating an entire new table.
- LILA no longer outputs a MySQL warning message, when the database already exists.
- Replaced python interrupt blabla with a Goodbye message and added some cleanup when LILA is ended.

0.4-beta

- First beta release. README, INSTALL, CHANGELOG, lila.py and lila.cfg is now included in the archive.

- Completely restructured DNS resolving: LILA now uses native python libraries instead of bash commands like grep or cut and works more efficiently. If resolver 1 and 2 get the same hostname it will be displayed only once.
- LILA determines the terminal type (linux / xterm) and sets color code escape sequences accordingly. This fixes an issue where colors were lacking on a standard linux console.
- SSH support: It's now possible to run LILA on your (fast) main computer, while DNS resolving is performed via SSH on the firewall computer. The MySQL database is stored on the local (faster) computer. This results in a speedup, especially when your main computer is faster than your firewall, what is usually the case.
- LILA now better detects the path of the config file lila.cfg.
- It's now possible to select individual color rules for each chain in the config file.
- LILA now creates the database automatically if it doesn't exist yet.
- Some minor code changes.

0.3-alpha

- The WHOLE iptables live log file is stored into a database before LILA starts the first time. Initalizing will take some time, but searches for the last n logs are a lot faster, especially for large values of n. New logs since the last start are automatically detected and added to the database. (This might result in performance problems for very large files and/or very slow computers).
- LILA can now list newly added (unviewed) entries since its last start (command line option -c)
- The DNS and the current live log table can now be deleted within the manage tables interface.
- Fixed an SQL error, when option –nodns was used.
- Created configuration file for lila for easier setup. (lila.cfg must exist)
- It's now possible to list all previous entries using command line option -n ALL
- Restructured the way LILA outputs colors. (module color.py deprecated)
- Some minor code changes.

0.2-alpha

- Added usage() and help screen. Type lila –help or lila -? to view.
- When listing the last n entries, dupes within the last t seconds won't be displayed, thus resulting in less than n entries. If you want LILA to completely ignore dupes and print every IP entry just once, you can define a very high time interval using the command line option -t [sec]. Command line option -s always shows every entry.
- DNS resolving and output can be disabled via command line option.
- LILA now creates the current working table automatically on startup if it doesn't exist yet.
- Use of hashlib instead of deprecated python module md5.
- Replaced standard table name current everywhere with a variable containing the name of the table, which is defined in the init section.
- It's now possible to view and delete the staticfile SQL tables created by LILA. (command line option -m).
- Some minor code changes.

0.1-alpha

- Initial alpha release.

6 Final notes, contact information

Like stated in the description section: If there are any remaining questions, comments or anything else please feel free to contact me at any time. It's always nice to get feedback :-)

Author:	`Joachim Fix`
Email:	`jfix@lavabit.com`
WWW:	`https://sourceforge.net/projects/lila/`
License:	`GNU GPLv3`