

The systraq Manual

Werurwe 2026

Joost van Baal-Ilic

This document is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this manual (see COPYING); if not, check with <http://www.gnu.org/copyleft/gpl.html> (<http://www.gnu.org/copyleft/gpl.html>) or write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA.

Table of Contents

1. Introduction.....	1
1.1. What is it?	2
1.2. Why?.....	2
2. Installation.....	3
2.1. Requirements.....	3
2.2. Configure the package (and optionally rebuild documentation).....	3
2.3. Install scripts and documentation	4
2.4. User account.....	4
2.5. Set up configuration files.....	4
2.6. Inspecting current state of your system, making the first snapshot.....	6
2.7. Setting up cronjob.....	7
3. Daily Maintenance.....	7
4. Internals.....	8
4.1. Files used	8
4.2. Dependencies.....	9
4.3. The systraq scripts.....	9
5. Hacking on systraq	9
5.1. Version Control	9
5.2. Generating .tar.gz from version control	9
6. Contact, other tools.....	10
6.1. Contact information, reporting bugs.....	10
6.2. Similar tools.....	10

1. Introduction

1.1. What is it?

systraq Is a small set of simple scripts monitoring your system, and warning you when system files change. systraq Daily sends you an email listing the state of your system. Furthermore, if critical files have changed, you'll get an email within a shorter notice. It consists of few very small shell scripts. It's written for Debian GNU/Linux, but very likely runs on any Unix like operating system. (Examples and default configuration will likely be somewhat Debian centered, 'though.)

It can help you keeping your system secure. However, in case you're planning to use systraq on a multi user system, make sure you really want to do the monitoring this script offers: it might not comply with your site's privacy policy. Getting informed when users' config file change might be too intrusive.

1.2. Why?

In the early 2000s, I had some *BSD boxes, which regularly mail stuff like:

```
checking setuid files and devices:
```

```
checking for uids of 0:
root 0
toor 0
```

```
checking for passwordless accounts:
```

```
hille.mdcc.cx login failures:
Oct 11 11:31:52 hille login: 1 LOGIN FAILURE ON ttyv0
Oct 11 11:31:52 hille login: 1 LOGIN FAILURE ON ttyv0, .^[^[^[MS
```

```
hille.mdcc.cx refused connections:
```

```
Disk status:
```

Filesystem	1K-blocks	Used	Avail	Capacity	Mounted on
/dev/ad0s1a	39647	27927	8549	77%	/
/dev/ad0s1f	1016303	857921	77078	92%	/usr
/dev/ad0s1g	7026508	6219148	245240	96%	/usr/home
/dev/ad0s1e	19815	6712	11518	37%	/var
procfs	4	4	0	100%	/proc

```
Network interface status:
```

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lp0*	1500	<Link#1>		0	0	0	0	0
ed0	1500	<Link#2>	00:00:e8:6b:a9:3b	651828	1177	4029190	4	36522

```
=====
```

```
/etc/sshd_config diffs (OLD < > NEW)
```

```
=====
```

```
1c1,11
```

```

< #      $OpenBSD: mailer.conf,v 1.3 2000/04/06 18:24:19 millert Exp $
---
> # This is ssh server systemwide configuration file.
>
> Port 22
> #Protocol 2,1

```

On OpenBSD boxes, the shellscreens `/etc/daily`, `/etc/weekly` and `/etc/monthly` kick off the process to generate these status mails. The shellscreen `/etc/security` is called, as well as the `mtree(8)` (<http://www.tac.eu.org/cgi-bin/man-cgi?mtree+8+NetBSD-1.5.1>) command.

I very much like this system, taking care of the automatic monitoring of my system. I run GNU/Linux also, these boxes lacked such a system (tripwire is too heavyweigh for my demands.) This system seemed not very portable to GNU/Linux, unfortunately. (Which is another way of stating: I'm too lazy to port the complete system.)

Jeremy Weatherford (<http://xidus.xidus.net/>) wrote FileTraq (<http://filetraq.xidus.net/>) for his Red Hat Linux box. This small tool could be regarded as a first estimate to what I wanted. Jeremy runs it as root, I believe. I want to avoid that as much as possible. I do want to monitor files like `/etc/shadow`, but I do not want to get the diff emailed when these change. I *do* want to get a notice if such a file changes. Christoph Lameter (<http://lameter.com/>)'s `debsums` (<http://packages.debian.org/stable/admin/debsums.html>) is a tool, for monitoring files installed from Debian packages, which has functionality like this. On Debian systems, there's `checksecurity(8)` in the `cron` package, which monitors permissions on device files.

So, I mixed ideas of the BSD 'daily run output' style emails with FileTraq and some other tools. That's how systraq came to be.

2. Installation

2.1. Requirements

You might need the GNU version of utilities like `cut` and `ls`: I've only tested sytraq on a GNU/Linux system. The systraq tool works nice with the Debian `debsums` package; however, systraq is useful too on systems lacking this package.

You need Jeremy Weatherford's (<http://xidus.xidus.net/>) FileTraq (<http://filetraq.xidus.net/>). However, beware! Jeremy no longer seems to maintain FileTraq. You'll need an up to date version; the Debian package `filetraq` `>= 0.2-10` by Sergio Talens-Oliag is fine. If you are on a Debian system, you know how to get this. If you are on another system, you can get the Debian `filetraq` version from the master ftp site (<ftp://ftp.debian.org/debian/pool/main/f/filetraq/>) or from any other mirror (<http://www.debian.org/mirror/list>). Be sure to at least get the files `filetraq_0.2.orig.tar.gz` and `filetraq_0.2-10.diff.gz` (or a later version). You can apply the diff with any **patch**.

2.2. Configure the package (and optionally rebuild documentation)

Run

```
$ ./configure
$ make
```

This will configure the package (you might want to pass some arguments to **./configure**, see the `INSTALL` file). You won't need typesetting tools: the typesetted documentation is shipped with the tarball. If you want to hack on the documentation, you'll need to have the documentation build tools installed. After editing documentation sources, just running **./configure** and **make** will be enough to get typesetting from the DocBook XML sources done.

2.3. Install scripts and documentation

(If you're upgrading from an old systraq installation, back up your configuration files in `/etc/systraq` and run

```
# make uninstall
```

from within your old unpacked tarball.) Run

```
# make install
```

This will install `st_snapshot`, `st_snapshot.hourly` and `systraq` in `/usr/sbin`. Furthermore, it will install documentation in `/usr/share/doc/systraq`. (Sample) configuration files will get installed in `/etc/systraq` and `/usr/share/doc/systraq/examples`. Helper scripts will be installed in `/usr/share/systraq`.

2.4. User account

Create a dedicated systraq user account. E.g.

```
# adduser --system --home /var/lib/systraq --disabled-password --force-badname _systraq
```

This user will read worldreadable files, and write files under `/var/lib/systraq`. Cronjobs will get run as this user, you might want to create a `~_systraq/.forward` (or whatever your MTA uses), to get these job's output in your mailbox.

2.5. Set up configuration files

2.5.1. Introduction

Example configuration files are distributed with this manual. (On Debian systems, the examples could be used as reasonable defaults, except for the `filetraq.conf` file, which needs to be generated for your particular system.) All configuration files are line oriented, lines with a leading `#` are ignored. We give some descriptions here.

2.5.2. `filetraq.conf`

The files listed in `/etc/systraq/filetraq.conf` will be checked by **filetraq** for changes in content every half hour. Diff's will be emailed to the administrator. The files `snapshot_pub.stat` and `snapshot_root.stat` should be listed here, as well as `systraq.sums` (all these files reside in `/var/lib/systraq`). It is advisable to also list every worldreadable file under `/etc/` (and possibly `/usr/local/etc/`) here. You also might like to list each user's `~/.ssh/authorized_keys` here.

All files listed in `filetraq.conf` should exist on your system, and should be worldreadable. (You can monitor non-world readable files in `/etc/` by adding them to `snapshot_root.list`).

You could create `filetraq.conf` using this Makefile:

```
filetraq.main.conf:
    echo '# $@: automatically generated' > $@
    find /etc /home/*/.ssh/authorized_keys* -perm -a+r -type f | \
        sort >> $@

filetraq.conf: filetraq.main.conf filetraq.tail.conf
    echo '# $@: generated from $^' | \
        cat - filetraq.main.conf filetraq.tail.conf > $@

.PHONY: filetraq.main.conf
```

where `filetraq.tail.conf` is

```
#
/etc/systraq/snapshot_pub.list
/etc/systraq/snapshot_pub.homelist
/etc/systraq/snapshot_root.list
/etc/systraq/snapshot_root.homelist
/etc/systraq/filetraq.conf
#
/var/lib/systraq/snapshot_pub.stat
/var/lib/systraq/snapshot_root.stat
/var/lib/systraq/systraq.sums
#
```

; that might get something useful, as a starter. Be sure to inspect `/usr/share/doc/systraq/examples/filetraq.conf` too.

You could install the Makefile in `/etc/systraq`.

If you don't like filetraq's default diff style, but, like me, prefer unified diff, do

```
# rm -f /etc/default/filetraq
# ln -s /etc/systraq/filetraq.default /etc/default/filetraq
```

2.5.3. `st_snapshot`'s file lists

Daily, **`st_snapshot`** will check all files as listed in its configuration files, aka listfiles. These listfiles are `/etc/systraq/snapshot_{pub,root}.list` and `/etc/systraq/snapshot_{pub,root}.homelist`. Example versions of these files are installed in `/usr/share/doc/systraq/examples` when running **`make install`**. The names of these files are given as the two arguments of **`st_snapshot`**.

`snapshot_pub.list` Should contain all world readable files for which we want to monitor existence, ownership, permissions and changes in content. It should contain `/var/lib/systraq/systraq.sums` too. `snapshot_root.list` should contain all files which are not world readable, we want to monitor. `snapshot_{pub,root}.homelist` should contain files we expect to find in homedirectories of users. All users homedirectories are scanned for files listed in these two listfiles. Think of files like shell startup scripts and stuff in `~/.ssh/` and `~/.rhosts`. You might want to add `.gnupg/revoke.asc` and `.gnupg/secring.gpg` too.

If a file listed in a listfile is a directory, all files residing in this directory, or any subdirectory thereof, gets counted in. Shell wildcards are allowed in the listfiles.

Now create the files `/etc/systraq/snapshot_{pub,root}.list` and `/etc/systraq/snapshot_{pub,root}.homelist`, using the example files. Once you're happy with the files, follow the instructions in the next section.

2.6. Inspecting current state of your system, making the first snapshot

Inspect all files listed in the listfiles, and decide whether their content is OK for your securitypolicy. Especially, the `authorized_keys` files need inspection. Once you're happy with their contents, create the `/var/lib/systraq` directory, and make sure the `_systraq` user can write to it. NB: if your copy of the `systraq` package uses another useraccount (the Debian package uses account "debian-systraq", e.g.) *that* user should have write-access instead, of course. Then, run **`st_snapshot`** manually:

```
# su -s /bin/sh _systraq
$ ST_OPHOMES=yes st_snapshot /etc/systraq/snapshot_pub.list \
  /etc/systraq/snapshot_pub.homelist > /var/lib/systraq/snapshot_pub.stat
```

Of course, if your copy of the `systraq` package uses another useraccount you should `su` to that user instead. And now, run as root:

```
# st_snapshot /etc/systraq/snapshot_root.list \
```

```
/etc/systraq/snapshot_root.homelist > /var/lib/systraq/snapshot_root.stat
```

Inspect the permissions as listed in the output files, and decide whether you're happy with them. Check if all files listed should really be on your system. (One could argue about whether one should have `~/.netrc`, `~/.rhosts`, `~/.ssh/identity`, `~/.shosts`, `/etc/exports`, `/etc/*hosts.equiv` . Of course, this depends on your planned use of the system.) If you're not happy, fix the permissions and ownerships. You might like to take a look at the OpenBSD `/etc/security` script (<http://www.openbsd.org/cgi-bin/cvsweb/src/etc/security?rev=1.49&content-type=text/x-cvsweb-markup>) to get inspiration.

Make sure you trust all binary files, which are not in vendor-supplied packages (e.g. stuff in `/usr/local/bin/` on Debian systems), as they are on your system now. (You could e.g. reinstall them from trusted sources.) Once your system is considered safe, generate a file containing checksums of these files. You can generate this by running e.g., as user `_systraq`,

```
$ find /usr/local/sbin /usr/local/bin /usr/local/lib \
    /usr/local/share -type f -exec sha256sum {} + | \
    sort -k 2 >/var/lib/systraq/systraq.sums
```

Make sure you trust all files in your `filetraq.conf` file, and create the directory `/var/lib/systraq/filetraq/`. Then run, as user `_systraq`,

```
$ filetraq /etc/systraq/filetraq.conf \
    /var/lib/systraq/filetraq
```

to create the first `filetraq` backup.

2.7. Setting up cronjob

The `systraq-version/etc/systraq` file is installed as `/etc/cron.d/systraq`. (If your cron doesn't look in this directory (but *has* support for `cron.d` style directories), then activate it by doing

```
# ln -s /etc/cron.d/systraq /etc/cron.d/systraq
```

.) This makes sure **filetraq** gets run every half hour, **systraq** gets run daily, and the `systraq` status files get updated by running **st_snapshot**, using the **st_snapshot.hourly** wrapper, each hour.

3. Daily Maintenance

When a user is added to the system: update `filetraq.conf` with this user's `authorized_keys`.

`filetraq.conf` needs maintenance also once files listed there have been removed by system upgrades, or once files have been added to e.g. `/etc/`.

In case Debian packages are installed with missing `/var/lib/dpkg/info/*.md5sums` file, things break. Consult the examples section in the `debsums` manpage, for a hint on how to deal with these broken packages. Alternatively, one can do:

```
debsums -s > /tmp/sums 2>&1
grep 'no md5sums for' /tmp/sums | awk '{print $5}' > /tmp/pkgs
```

check the contents of `/tmp/pkgs`.

```
apt-get update
apt-get --reinstall install `cat /tmp/pkgs`
debsums --silent --generate=missing,keep `cat /tmp/pkgs`
apt-get clean
```

When installing or upgrade stuff in `/usr/local`, be sure to update `/var/lib/systraq/systraq.sum`s with the correct checksums.

4. Internals

4.1. Files used

We list all files used by the `systraq` system, along with a short description of their role.

Files used

`/var/lib/systraq`

homedir of `_systraq` user.

`/var/lib/systraq`

stores `systraq` status files, should be writable by `_systraq` user.

`/var/lib/systraq/snapshot_pub.stat`

`/var/lib/systraq/snapshot_root.stat`

stdout of **st_snapshot**, listing permissions, ownership and checksums of some files, both publicly readable, as well as non-world readable.

`/etc/systraq/filetraq.conf`

configuration file for **filetraq**, listing files to get monitored.


```
/etc/systraq/snapshot_pub.list  
/etc/systraq/snapshot_root.list  
/etc/systraq/snapshot_pub.homelist  
/etc/systraq/snapshot_root.homelist
```

configuration files for **st_snapshot**, listing both publicly readable, as well as non-world readable files to get monitored, as well as files to be found in homedirectories. Script **st_snapshot.hourly** passes these to **st_snapshot** in its two arguments.

```
/var/lib/systraq/systraq.sums
```

checksums of binary files not in Debian packages, verified by running **systraq**.

4.2. Dependencies

FIXME: diagram listing dependencies: what calls what, what reads and writes what.

4.3. The systraq scripts

The **systraq**, **st_snapshot** and **st_snapshot.hourly** scripts come with their own manpages, distributed with this manual.

5. Hacking on systraq

Some hints for those who'd like to hack the systraq software. This section is only interesting for software developers.

5.1. Version Control

Systraq is maintained using git on Debian's Alioth, in the collab-maint (<https://wiki.debian.org/Alioth/PackagingProject>) project. If you have access, you should be able to do:

```
$ git clone ssh+git://git.debian.org/git/collab-maint/systraq.git
```

If you don't have git write access, you can check <http://anonscm.debian.org/gitweb/?p=collab-maint/systraq.git> for readonly Web access; or do

```
$ git clone git://anonscm.debian.org/collab-maint/systraq.git
```

for access using your git client. Commit messages get sent to a Debian Package Tracking System (<https://packages.qa.debian.org/>) mailing list. See <https://www.debian.org/doc/manuals/developers-reference/resources.html#pkg-tracking-system> for more information.

5.2. Generating .tar.gz from version control

If you'd like rebuild documentation, e.g. for creating a new .tar.gz release, you'll need the tools listed below.

You need the SGML Declaration for XML 1.0 for building. On Debian systems, this is installed as `/usr/share/sgml/declaration/xml.dcl`. `./configure` will give you hints on what to do if it fails to find this file.

You need the DocBook XSL stylesheet (<http://docbook.sourceforge.net/projects/xsl/>) for manpages for building. On Debian systems, this is installed as `/usr/share/sgml/docbook/stylesheet/xsl/nwalsh/manpages/docbook.xsl`. `./configure` will give you hints on what to do if it fails to find this file.

You need an XSLT engine for building. By default, *systraq* tries to find `xsltproc` (<http://xmlsoft.org/XSLT/>) from Daniel Veillard. This package is shipped with many GNU/Linux and BSD distributions. Alternatively, *Saxon* (<http://saxon.sourceforge.net/>), written in Java, from Michael Kay or *XML::XSLT* (<http://xmlxslt.sourceforge.net/>), written in Perl, from Geert Josten, Egon Willighagen e.a. might work for you. However, you'd have to hack *systraq*'s `man/Makefile.am` to use these.

You need *OpenJade* (<http://openjade.sourceforge.net/>), as maintained by Castle, Clasen, Ibach, Martin, Nilsson e.a. to typeset this manual. *OpenJade* is shipped with many GNU/Linux and BSD distributions. Alternatively, you can use James Clark (<http://jclark.com/>)'s *Jade* (<http://jclark.com/jade/>) from <ftp://ftp.jclark.com/pub/jade/>. Beware: there hasn't been a *Jade* release between October 1998 (1.2.1) and October 2004. You might need Debian patches: The Debian package has had patches applied on 8 Jun 2004 and maybe later. So you're likely better off using *OpenJade*: upstream for this extension of *Jade* was alive at 2004-09. The `./configure` script will pick whatever is available on your system.

Furthermore, you need **jadetex** and **pdfjadetex** as shipped with *JadeTeX* (<http://jadetex.sourceforge.net/>) (and with many GNU/Linux and BSD distributions) to generate PostScript and PDF from this document.

You need TeX's **dvips** to generate PostScript from *JadeTeX*'s DVI. If it's not included with your system, get it from the TeX Live distribution (<http://www.tug.org/texlive/>).

Finally, you need *w3m* (<http://w3m.sourceforge.net/>) to convert HTML output to plain ascii. If `./configure` doesn't find *w3m* on your system, it'll use *Lynx* (<http://lynx.isc.org/release/>). Both text-browsers are shipped with many GNU/Linux and BSD distributions.

6. Contact, other tools

6.1. Contact information, reporting bugs

If you have any questions or remarks about *systraq*, you can mail the author at `<joostvb-systraq-20151105@mdcc.cx>`. You can also use this address for reporting bugs (reading the *How To Ask Questions The Smart Way* (<http://www.catb.org/~esr/faqs/smart-questions.html>) document before reporting might be useful) . However, if the bug you've found is present in the *systraq* Debian package too, please use the Debian Bug Tracking System (<http://www.debian.org/Bugs/Reporting>) for reporting.

6.2. Similar tools

I believe diffmon (<http://packages.debian.org/unstable/admin/diffmon.html>) does about the same as this tool.

FAM (<http://oss.sgi.com/projects/fam/>) (File Alteration Monitor) could be used by systraq (instead of cron): it is for a particular process to "subscribe" to changes to a file / directory. FAM then implements the system-dependent best way to do that (e.g. dnotify on modern Linux) and, if more than one process is interested in the same file, centralises the polling (if polling is necessary), so that less resources are taken in total.

Aide (<http://aide.sf.net/>) and Osiris (<http://osiris.shmoo.com/>) are big packages for use in sites where demands are high.