The systraq Manual

August 2004

Joost van Baal

Table of Contents

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, 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?

I have some *BSD boxes, which regularly mail me 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 shellscripts /etc/daily, /etc/weekly and /etc/monthly kick off the process to generate these status mails. The shellscript /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 heavyweigth 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 systraq.

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 (to be sure Debian bug #251010 is fixed). 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. Install scripts and documentation

Run

```
$ ./configure
$ make
# make install
```

This will install `st_snapshot` and `systraq` in `/usr/local/bin/`. Furthermore, it will install documentation in `/usr/local/share/doc/systraq/`. Configuration files will get installed in `/usr/local/etc/systraq/`.

2.3. User account

Create a dedicated systraq user account. E.g.

```
# adduser --system --home /usr/local/var/lib/systraq --disabled-password systraq
```

This user will read worldreadable files, and write files under `/usr/local/var/state/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.

2.4. Set up configuration files

2.4.1. 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.

2.4.2. filetraq.conf

The files listed in `/usr/local/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.md5sums` (all these files reside in `/usr/local/var/state/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` and `~/.ssh/authorized_keys2` file 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 -perm -a+r -type f | sort ; \
          ls -1 /home/*/.ssh/a* ) | sort >> $@

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

where filetraq.tail.conf is

```
#
/usr/local/etc/systraq/snapshot_pub.list
/usr/local/etc/systraq/snapshot_pub.homelist
/usr/local/etc/systraq/snapshot_root.list
/usr/local/etc/systraq/snapshot_root.homelist
/usr/local/etc/systraq/filetraq.conf
#
/usr/local/var/state/systraq/snapshot_pub.stat
```

```
/usr/local/var/state/systraq/snapshot_root.stat
/usr/local/var/state/systraq/systraq.md5sums
#
```

; that might get something useful, as a starter.

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

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

2.4.3. st_snapshots's file lists

Daily, **st_snapshot** will check all files as listed in its configuration files, aka listfiles. These listfiles are `/usr/local/etc/systraq/snapshot_{pub,root}.list` and `/usr/local/etc/systraq/snapshot_{pub,root}.homelist`. These files are installed when running **make install**.

`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 `/usr/local/var/state/systraq/systraq.md5sums` too. `snapshot_root.list` should contain all files which are not world readable, we wanna 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.

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

Inspect all files listed in the listfiles, and decide wether their content is ok for your securitypolicy. Especially, the `authorized_keys` files need inspection. Once you're happy with their contents, create the /usr/local/var/state/systraq/ directory, and make sure the systraq user can write to it. Then, run **st_snapshot** manually:

```
# su -s /bin/sh systraq
$ st_snapshot pub > /usr/local/var/state/systraq/snapshot_pub.stat
```

and, as root:

```
# st_snapshot root > /usr/local/var/state/systraq/snapshot_root.stat
```

Inspect the permissions as listed in the output files, and decide wether you're happy with them. Check if all files listed should really be on your system. (One could argue about wether 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 Debian packages (e.g. stuff in `/usr/local/bin/`), as they are on your system now. (You could e.g. reinstall them from trusted sources.) Once you feel safe, generate a file containing md5sums of these files. You can generate this by running e.g., as user systraq,

```
$ md5sum ‘find /usr/local/sbin/ /usr/local/bin/ /usr/local/lib/ \
    /usr/local/share/ -type f | sort‘ > \
    /usr/local/var/state/systraq/systraq.md5sums
```

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

```
$ filetraq /usr/local/etc/systraq/filetraq.conf \
    /usr/local/var/state/systraq/filetraq
```

to create the first filetraq backup.

2.6. Setting up cronjob

The `systraq-`*`version`*`/etc/systraq` file is installed as `/usr/local/etc/cron.d/systraq`. Activate it, by doing

```
# ln -s /usr/local/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** each hour.

3. Daily Maintenance

When a user is added to the system: update `filetraq.conf` with this user's `authorized_keys` (or `~/.ssh/authorized_keys2`).

`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 `/usr/local/var/state/systraq/systraq.md5sums` with the correct md5sums.

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

`/usr/local/var/lib/systraq/`

homedir of systraq user.

`/usr/local/var/state/systraq/`

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

`/usr/local/var/state/systraq/snapshot_pub.stat`
`/usr/local/var/state/systraq/snapshot_root.stat`

stdout of **st_snapshot**, listing permissions, ownership and md5sums of some files, both publicly readable, as well as non-world readable.

`/usr/local/etc/systraq/filetraq.conf`

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

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

configutation 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.

`/usr/local/var/state/systraq/systraq.md5sums`

md5sums 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 command

systraq runs various system commands, to inspect the state of the system: what is it doing now, what has it been doing recently, are we running to hardware limitations. It inspect some files in users' homedirectories, as well as some system files, for frequently seen flaws. E.g. unsafe umask setting in shell startup scripts, or unsafe PATH in these scripts. It runs **debsums**, to check md5sums as stated in packaging files with the sums of the actual files running the system. It runs **md5sum** to check md5sums as locally maintained in `/usr/local/var/state/systraq/systraq.md5sums`.

4.4. The st_snapshot command

The **st_snapshot** command comes with its own manpage, distributed with this manual.

5. Hacking on systraq

Systraq is maintained using CVS on topaz.conuropsis.org. If you have access, you should be able to do:

```
$ CVSROOT=:ext:topaz.conuropsis.org:/var/cvs cvs co ad1810.com/packages/systraq
$ less ad1810.com/packages/systraq/bootstrap
```

If you don't have CVS write access, you can check http://mdcc.cx/~joostvb/packages/systraq/ for a reasonable fresh copy of the CVS.

6. More information, other tools

There is a **st_snapshot** manpage. You can mail me at <`joostvb-systraq-20041015@mdcc.cx`>.

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 and Osiris are big packages for use in sites where demands are high.