

# Tripwire install doc for Solaris 2.6 / 7

Robin 30april2001

## Installation

1. Get the tripwire academic source release (ASR) v1.3.1 :

```
283196 Apr 25 18:40 Tripwire-1_3_1-1_tar.gz
```

2. Untar it and modify Makefile:

```
change MANDIR = /usr/local/man          # This needs to change to reflect the path
change INSTALL :
        INSTALL= /usr/sbin/install      # common
```

Change the usage of the \$(INSTALL) command, as the Solaris install command has a different syntax:

Old:

```
(cd configs; $(INSTALL) -m 444 tw.config $(DESTDIR))
```

New:

```
(cd configs; $(INSTALL) -f $(DESTDIR) -m 444 tw.config )
```

Also change the \$(INSTALL) line for tw.db\_TEST :

Old \$(INSTALL) -m 444 tests/tw.db\_TEST \$(DATADIR)

New: \$(INSTALL) -f \$(DATADIR) -m 444 tests/tw.db\_TEST

3. Modify src/Makefile and change the \$INSTALL syntax as above , so it now reads:

```
$(INSTALL) -f $(DESTDIR) -m 555 tripwire
```

```
$(INSTALL) -f $(DESTDIR) -m 555 siggen
```

4. Modify man/Makefile so it reads :

```
...
install:
# added mkdir as man5&8 may not exist -robin
    mkdir -p $(MANDIR)/man5 $(MANDIR)/man8
    cp siggen.8 $(MANDIR)/man8
...
```

5. Edit include/config.h and make sure it contains settings for Solaris .

6. Build :

```
make
make test
make install
```

Ideally this should be done on a freshly installed system which has never been connected to a public network, and you should do it in single-user mode locally on the box (not via the network), and then copy the tripwire binaries onto a disk which is physically read-only (e.g. a CD-ROM), but this isn't always practical: use your own judgement.

Now edit the tw.config file (in /usr/local/bin/tw) with suitable settings for your system. It should have some basic settings for SunOS 5.x in there already. Make sure you add

```
/usr/local/bin/tw          R
!/usr/local/bin/tw/databases
```

plus any other files or directories which are specific to your system.

6. Now do the first system scan to create the first database of file signatures:

```
cd /usr/local/bin/tw
./tripwire -initialize
```

this creates ./databases and puts tw.db\_hostname in there, so make sure you cd into /usr/local/bin/tw before running tripwire with the -initialize option

7. Now copy the generated database into /var/tripwire:

```
mv ./databases/tw.db_hostname /var/tripwire
```

8. to test it works , add a comment line to e.g. /etc/hosts and then run

```
/usr/local/bin/tw/tripwire
```

( you may want to append '> /tmp/trip.out 2>&1' to the above command if there is a lot of output )

It should report the change you made (as well as possibly many others it has found if you are running on a live system)  
It takes about 1-2 minutes to scan 8000 files on an ultra-5 (e.g. hazel) and 8 min to scan 55000 files on an E420 (apple).

You may have to edit the tw.config file to exclude files you don't want reported on, and run tripwire -initialize again.  
N.B. After each time you run -initialize you will have to copy the database from ./databases to /var/tripwire

## Using tripwire regularly

You should run tripwire daily (or as often as you like) from cron.

When a file changes (e.g. you add an entry to /etc/hosts) and you want to make that change permanent, run either

```
cd /usr/local/bin/tw
./tripwire -interactive to selectively update the database.
```

N.B. cd to /usr/local/bin/tw first, as it always writes to ./database

answer y to each question it asks (or Y to update all the files it finds which have changed for that line)

e.g.

```
/etc/hosts
st_mtime: Wed May 5 15:30:37 1993 Wed May 5 15:24:09 1993
st_ctime: Wed May 5 15:30:37 1993 Wed May 5 15:24:09 1993
--> File: '/etc/hosts
--> Update entry? [YN(y)nh?] y
```

or, if there is just one file you want to update:

```
cd /usr/local/bin/tw
tripwire -update /etc/hosts
```

NB After each time you run -update or -interactive you will have to copy (or mv) the database from ./databases to /var/tripwire

## Updating the tw.config file on all hosts

Whenever you change the tw.config file you should copy it out to all hosts and update their databases: there is a script to do this, which may come in handy :

```
apple:/usr/local/scripts/tripw_update_config
```

## Making Tripwire run faster

Tripwire allows you to selectively skip certain signatures at run-time through a command-line option. For example, if you wish to run Tripwire on an hourly basis, even performing only MD5 checks might be computationally prohibitive. For this application, checking only the CRC32 signature might be desirable. To do this, assuming that only MD5, Snefru, and CRC32 were used when the database was initialized, you would type:

```
tripwire -i 1 -i 2
```

This tells tripwire to ignore signature 1 and signature 2. Furthermore, for daily Tripwire runs, you could specify using only MD5 and CRC32. Finally, for weekly runs, you could run Tripwire with all three signatures.

To find added or deleted files, with no signature checking, use:

```
tripwire -i all
```