Tripwire Configuration

https://fcerbell.github.io/Debian113Server100Tripwiretodetectpenetration-en/

```
apt-get install -y tripwire
```

Disable false positive checks

I disable monitoring on some folder structures, including the `/root/` folder, but I force the check of some exceptions such as `/root/bashrc` and `/root/bash_history`. I also change the policy for the log files, because they can have an inode change when LogRotate rotates them.

```
sed -i 's~/etc/rc.boot~#&~' /etc/tripwire/twpol.txt
sed -i 's~/root/~#&~' /etc/tripwire/twpol.txt
sed -i 's~/proc~#&~' /etc/tripwire/twpol.txt
# Ignore inode change because of logrotate
sed -i 's~/var/log[^;]*~&-i ~' /etc/tripwire/twpol.txt
sed -i 's~#\(/root/.bashrc.*\)~\1~' /etc/tripwire/twpol.txt
sed -i 's~#\(/root/.bash_history.*\)~\1~' /etc/tripwire/twpol.txt
```

Notifications

Use a mail command instead of a direct SMTP connection (there is currently no real MTA on the server)

```
sed -i 's~^MAILMETHOD.*=.*~MAILMETHOD =SENDMAIL~' /etc/tripwire/twcfg.txt
echo 'MAILPROGRAM=/usr/sbin/sendmail -oi -t' >> /etc/tripwire/twcfg.txt
echo 'GLOBALEMAIL = root' >> /etc/tripwire/twcfg.txt
```

Compile and sign the configuration file

This step freezes the TripWire configuration file by compiling it and signing it with the *site-key*.

```
/usr/sbin/twadmin --create-cfgfile -S /etc/tripwire/site.key
/etc/tripwire/twcfg.txt
```

Compile and sign the policies file

This step protects the policy definition file by compiling it and signing it with the *site-key*.

```
/usr/sbin/twadmin --create-polfile -S /etc/tripwire/site.key
/etc/tripwire/twpol.txt
```

Initialize the database

Let’s ask TripWire to take a snapshot of the current system files status, write it in a database and sign the snapshot with the *local-key*.

```
ln -s site.key router-local.key
```

```
/usr/sbin/tripwire --init
```

Test runs

Check and notify

This asks TripWire to check that all the current system files were not changed against the defined policies. We should have very little (to no) change, as we just built the database. The results will be compiled in a report that will be written to the disk and sent by email. This is not only a check test, but also an email alert test.

```
/usr/sbin/tripwire --check --email-report
```

Update and sign the database according to the last check/report

Once I receive an email alert from Tripwire informing me that there were unauthorized changes, I can investigate and finally *validate* the report and update the database according to the detected changes. This is also an administration command. Tripwire will ask for the *local-key* to update the database.

```
/usr/sbin/tripwire --update -a -r /var/lib/tripwire/report/`ls -rt
/var/lib/tripwire/report/ | tail -n 1`
```

Interactive update and sign

When I make some system changes, such as an installation, a change in a configuration file, … I already know that Tripwire will complain and I don’t want to wait for the email report, thus I can acknowledge the changes immediately. This command triggers a check and opens the report in the text editor. In the report, every change is flagged as *normal* with `[x]`. I can reject some of them, and save/exit. Tripwire will ask for the *local-key* to update the database.

```
tripwire --check --interactive --visual vi
```