Mise en œuvre de TRIPWIRE

le 20 juin 2000

Patrick JUEN

Tripwire est un vérificateur d'intégrité de fichiers. Il permet de détecter les changements non autorisés dans les fichiers que vous aurez défini.
La surveillance peut être faite au niveau du contenu des fichiers ou des accès a ceux-ci. Tripwire exploite, pour ce faire, une base de données interne dans laquelle il gère des signatures associées à chaque fichier et répertoire référencés.

Installation de tripwire

Tripwire version 1.2 peut être récupéré sur :
- http://www.urec.cnrs.fr/securite/outils

Tripwire version 1.3 peut être récupéré, via un enregistrement de votre adresse mail, sur :
- http://www.tripwiresecurity.com

Dans le fichier Ported vous trouverez les paramétrages à mettre manuellement dans le ./Makefile.

Dans le fichier **./include/configs.h** mettre le nom du fichier **./config/conf-<os>.h** qui correspond à votre operating system.

ATTENTION: pour linux, préférer conf-svr4.h a conf-linux.h

Dans le fichier ./include/configs.h modifier le chemin par défaut ou se trouve le fichier de configuration nommé tw.config. (celui qui indique les fichiers à surveiller) via:
#define CONFIG_PATH "/tmp/genek"

Dans le fichier ./include/configs.h modifier le chemin ou se trouve la base
#define DATABASE_PATH "/tmp/genek"

Il est fortement conseillé de mettre les fichiers de la configuration et de la base sur le disque d'une autre machine montée en lecture seule.
Pour sécuriser un parc de machines il peut exister un fichier de configuration et plusieurs bases de données. Chaque machine possède sa propre base de données. Elles sont créees et mises à jour par tripwire.

Créer le fichier de configuration **tw.config** dans le répertoire CONFIG_PATH

Syntaxe du tw.config

```
/bin                #Vérification du répertoire bin et des sous répertoires. Tripwire
                    est récursif si la structure de fichier est continue.

!/tmp               #Ne pas surveiller tmp et ses enfants.
=/tmp               #Ne pas surveiller tmp mais le faire pour ses enfants.
/etc/inetd.conf     #vérifier le fichier inetd.conf


/bin    ma_verif    # Vérifier bin en utilisant la règle: ma_verif
```

Ecriture des règles de vérifications

@@define nom_de_la_regle [[+|-][pinugsam123456789] ...]

- ignorer les attributs suivants

+ vérifier les attributs suivants

p permission and file mode bits

i inode number

n number of links (i.e., inode refer ence count)

u user id of owner

g group id of owner

s size of file

a access timestamp

m modification timestamp

c inode creation/modification timestamp

Les cryptages possibles

0	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***
Null !	Md5	Snefru	Crc32	Crc16	Md4	Md2	Sha	Haval	Null !
!	70K/S	31K/S	110K/S	130K/S	332K/S	3K/S	100K/S	100 K/S	!

Pour la vitesse, privilégier MD4 qui est fiable mais craquable.
Pour la fiabilité, privilégier MD5 qui est plus lent mais plus sûr.

Les règles prédéfinies

R [R]ead-only (+pinugsm12-ac3456789) **(règle par défaut)**

L [L]og file (+pinug-sacm123456789)

N ignore [N]othing (+pinusgsamc123456789)

E ignore [E]verything (-pinusgsamc123456789)

> taille de fichier monotone (+pinug>-samc1233456789). Les changements seront ignorés si le fichier diminue de taille. Utile pour les fichiers de log qui ne font que grossir.

Penser à protéger le(s) fichier(s) lançant tripwire et l'exécutable de tripwire.
Ne pas laisser les sources de tripwire sur la machine à tester.
Lancer automatiquement tripwire après un reboot.

Exemple

```
#Exemple de fichier tw_config

@@define    REGLE       +pinug5-cas012346789

#@@undef   REGLE       Pour vous monter la syntaxe

@@ifdef      HPUX
@@ define REGLE+pinug-cas0123456789
@@ifndef     HPUX

@@ifhost     cristallo.polycnrs-gre.fr
@@ define REGLE+pinug-cas0123456789
@@else
@@ define REGLE+pinug012-cas3456789
@@endif

/bin     REGLE
/sbin   REGLE
/dev    REGLE
/lib      REGLE
/etc     REGLE
```

Compilation

```
./make
./make install
```

Utilisation de tripwire

Lancer la création de la base

```
./src/tripwire -initialize
```

Suivant les cas cela peut durer 15 minutes sur un Pentium 200Mhz en linux 6.2 avec la vérification de tous les fichiers en Md5!.
En linux, n'utilisez pas l'exemple fourni (tw.conf.linux), il ne marche pas (core dump!).

Transférer le fichier de la base de donnée nouvellement créee vers sa destination définitive (DATABASE_PATH).

```
cp   /.src./databases/tw.db_<nom de l'ordi>   /tmp/genek
```

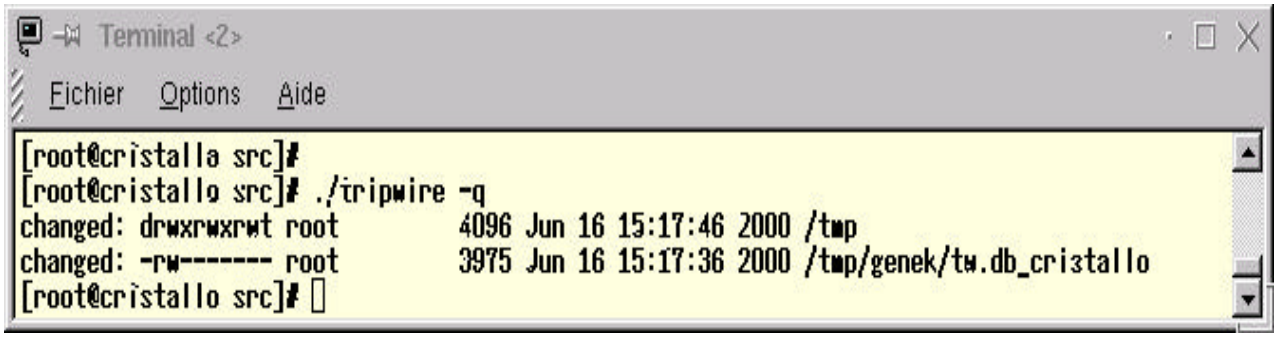
Pour vérifier l'intégrité du système, il suffit de faire:

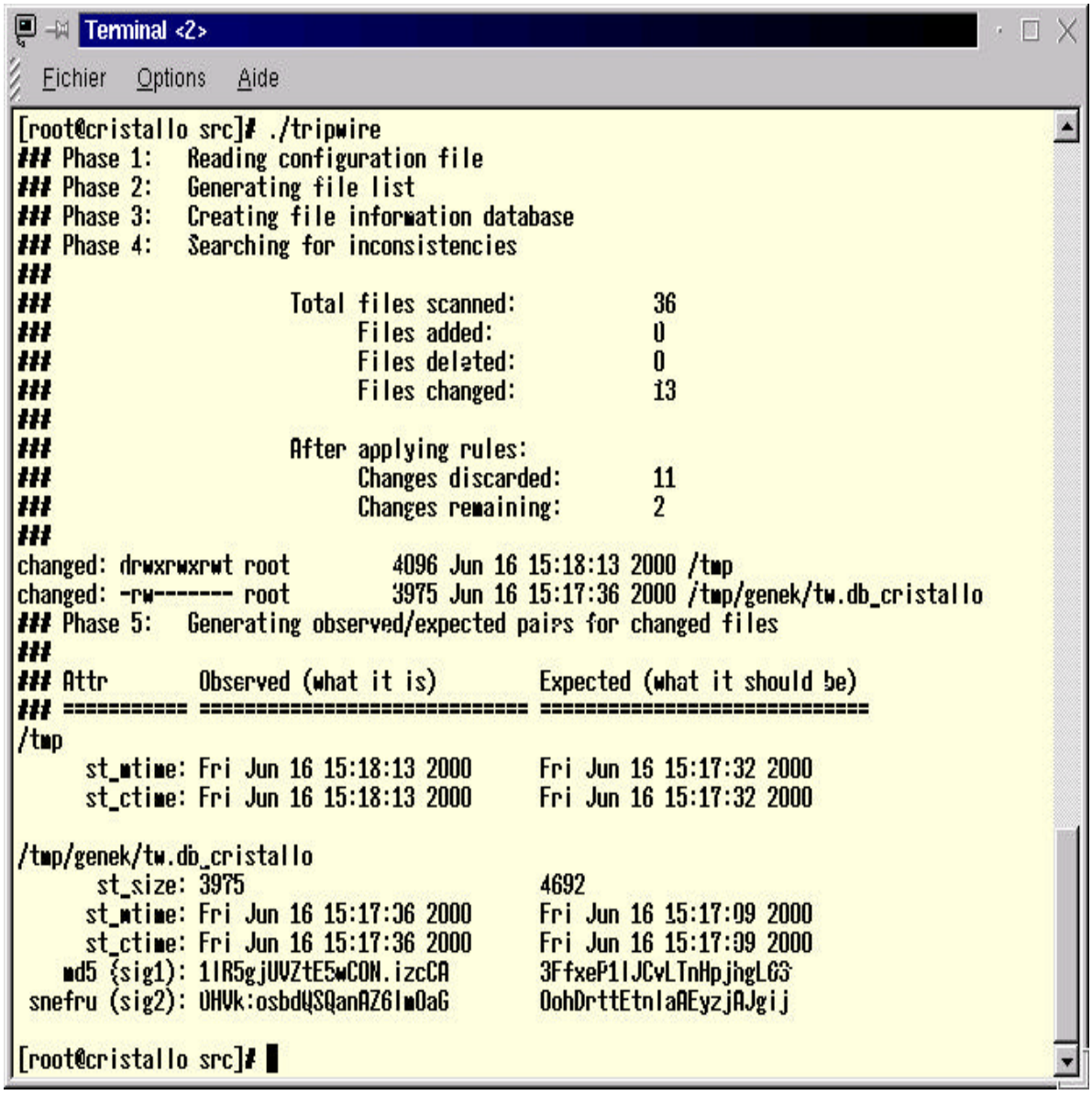
```
./src/tripwire
```

Options utiles:

-q	permet de réduire la verbose
-losedir	permet de ne pas donner d'informations sur les répertoires.
-interactive	permet de fonctionner en interactive pour valider les changements
-update toto	pour valider le changement d'un fichier nommé toto

Exemples de sortie





Utilitaire

Utilitaire pour voir les signatures d'un fichier

siggen nom_du_fichier

Utilitaire pour vérifier la base de données

twdb_check.pl