Un produit de contrôle d'intégrité de système UNIX : Tripwire

Gilles Plançon, plancon@idris.fr

PRINCIPE

Ce produit travaille à l'aide d'un fichier de configuration qui va contenir des binômes de la forme :

NOM DE RESSOURCE À SURVEILLER CRITÈRES DE SURVEILLANCE

Chaque ressource peut avoir son propre masque de critères de surveillance.
On peut contrôler l'intégrité de :

- une arborescence complète de fichiers;
- une partie seulement de cette arborescence;
- un simple élément de cette arborescence.

Les critères de surveillance sont :

- tous les champs de l'INODE – uid gid inode number .....;
- le contenu du fichier lui-même par génération de sceau à l'aide d'algorithmes de chiffrement tels que : MD2-MD4-MD5-Snefru-sha .....

On pourra calculer plusieurs signatures pour une même ressource si on estime qu'elle est très sensible. Mais attention il faut bien voir que :

- plus une signature est courte moins elle est fiable;
- plus une signature est élaborée, plus elle est sure, MAIS, en contrepartie de cette fiabilité accrue elle consomme plus de CPU pour être calculée.

Donc il faut bien faire attention aux éléments suivants :

- le nombre de ressources sur lesquelles on calcule une ou des signatures;
- la puissance de votre machine;
- la taille des fichiers sur lesquels on va calculer cette signature.

Toutes ces informations calculées ou recherchées par Tripwire seront mises dans un fichier qu'il appelle sa "Database". C'est en fait un fichier texte standard Unix avec des informations codées et d'autres en clair. Dans ce fichier il y aura 1 enregistrement par ressource surveillée. C'est ce fichier qui servira de référence pour trouver les différences entre 2 passages de Tripwire.

FONCTIONNEMENT

Tripwire a 4 modes de fonctionnement. Chacun de ces modes correspond à une fonction particulière.

- *Initialization Mode* : à utiliser au premier appel pour initialiser la *database*.
- *Checking Mode* : mode de surveillance qui ne modifie pas la *database*. Donne simplement entre 2 passages les fichiers supprimés-ajoutés-modifiés.
- *Updating Mode* : met à jour la *database* sélectivement sans la régénérer.
- *Interactive Mode* : demande à l'administrateur de valider les différences et régénère une nouvelle *database* de référence.

Dans ces 4 modes de fonctionnement Tripwire travaille en 5 phases d'exécution

- lecture du fichier de configuration qui contient les entrées à traiter pour en faire une analyse syntaxique;
- génération d'une liste des entrées à traiter;
- génération d'une *database* temporaire pour les entrées de la phase précédente;
- recherche des différences entre cette *database* et la *database* de référence créée à l'exécution précédente de Tripwire;
- génération d'un rapport d'anomalies pour toutes les différences trouvées entre les 2 *databases*.

LE FICHIER DE CONFIGURATION

C'est le point stratégique de ce produit. C'est lui qui va contenir les entrées, fichiers ou répertoires à surveiller. Pour déterminer les critères de surveillance on associera à chaque entrée un masque de traitement. C'est ce masque qui va contenir les options que l'on aura choisi de surveiller. Donc on y trouvera les champs de l'inode à vérifier ainsi que le ou les types de signatures à appliquer au fichier si on a choisi cette stratégie.

On a également la possibilité de faire précéder le nom de l'entrée d'un attribut de traitement particulier. Ce préfixe va indiquer si on traite ou non ou partiellement cette entrée.

Le nom du fichier de configuration est codé en dur dans l'exécutable Tripwire pour des raisons de sécurité. On paramètrera ce nom dans un fichier `config.h` qui sera utilisé à la compilation.

Un certain nombre d'exemples de fichiers de configuration pour différents types de plateformes sont livrés avec le produit. Il faudra quand même les adapter à son environnement.

LA DATABASE

C'est un fichier texte Unix. Il va contenir 1 enregistrement par entrée traitée conformément au fichier de configuration.

Comme pour le fichier de configuration son nom est codé en dur dans l'exécutable Tripwire, pour la même raison. Il se trouvera aussi dans le fichier `config.h` qui sera pris en compte à la compilation.

Ce fichier comporte toujours 21 champs par entrée. Les champs sont soit validés s'ils font partie du masque des critères à surveiller, soit à 0 s'ils ne sont pas dans le masque. La structure est la suivante :

- on trouvera d'abord le nom de l'entrée;
- puis 2 champs propres à Tripwire;
- puis 9 champs pour les différentes valeurs de l'inode que l'on peut surveiller;
- puis 10 champs pour les différents types de signatures que l'on peut calculer et qui sont toutes proposées en standard dans Tripwire.

Cette *database* peut devenir rapidement assez importante en taille, si l'on veut surveiller beaucoup de fichiers et/ou répertoires avec beaucoup d'attributs et de signatures.

D'où l'importance de bien coder son fichier de configuration.

RAPPORT D'EXÉCUTION

Dans ses 4 modes de fonctionnement Tripwire donne un rapport d'exécution. Dans ce rapport on trouvera :

- le nom des fichiers qui ont été détruits depuis la dernière exécution de Tripwire;
- le nom des fichiers créés depuis la dernière exécution de Tripwire;
- le nom des fichiers pour lesquels il y a eu une ou des modifications. Soit au point de vue inode, soit au point de vue du contenu du fichier. Dans le cas de modifications le rapport donne :
  - la valeur actuelle du ou des champs modifiés;

- la valeur qui se trouve dans la *database* de référence de la dernière exécution de Tripwire pour tous les champs modifiés.

Ce rapport d'exécution est très important et c'est lui qu'il faudra analyser avec beaucoup de soins à chaque exécution de produit Tripwire.

DIVERS

Faire tourner ce produit régulièrement en *Checking Mode*, dans un `cron` par exemple. Puis quand le nombre de modifications le justifie faire une exécution en *Interactive Mode* pour refaire une *database* à jour qui reflète aussi bien que possible l'état du système.

Ce produit n'a pas besoin de privilège particulier. Il faut seulement le mettre dans un répertoire dont l'accès est très limité, et d'autorisation d'exécution aussi limitée.

Pour que ce produit soit efficace à 100% il faudrait que le premier passage d'exécution de Tripwire se fasse sur un système qui vient d'être généré, et non encore connecté au réseau. Ceci pour minimiser au maximum les chances d'avoir un système qui a déjà été pollué avec un cheval de Troie par exemple.

Ce produit tourne sur un grand nombre de plateformes. Dans sa distribution il y a un fichier PORTED qui indique les spécificités de chaque machine avec les modifications à faire dans le `makefile` pour installer ce produit sans aucun problème.

Vous trouverez ce produit sur le serveur ftp de :

- `coast.cs.purdue.edu`
- dans `/pub/tools/unix/Tripwire`

CONCLUSION

Ce produit apparaît comme un bon complément aux autres outils de sécurité du domaine public tels que : COPS TIGER CRACK SKEY ANLPASSWD ..... .

Il est facile à mettre en œuvre, ne demande pas trop de ressources, et donne des indications précieuses sur les modifications apportées au système au fur et à mesure de sa vie.

Il est très facilement adaptable à son environnement et modifiable grâce à son fichier de configuration.