iwatch

Monitoring de système de fichier en temps réel utilisant inotify

iWatch est un outil Perl pour inotify pour superviser les changements dans des répertoires et fichiers spécifiques, en envoyant des alarmes à l'administrateur système en temps réel. Il peut :

- Envoyer des notifications via des email sur les changements
- Exécuter les actions programmables immédiatement
- Agir comme un HIDS (Host-based Intrusion Detection System) ou un vérificateur d'intégrité

iWatch peut être lancé comme service pour comme simple commande. Le mode service utilise un fichier de configuration xml. Le mode ligne de commande n'utilise pas de fichier de configuration.

Dans le fichier de configuration, chaque cible peut avoir son propre email de contact.

Options pour le mode service

-d Mode service
-f Spécifie le fichier de configuration. Défaut : /etc/iwatch/iwatch.xml
-p Fichier pid à utiliser. Défaut : /var/run/iwatch.pid
-v mode verbeux

Options pour le mode ligne de commande

-c Spécifie la commande à exécuter
-C Spécifier l'encodage. Défaut : utf-8
Spécifie une liste d'événements à monitorer
-m Adresse email de contact
-r Recherche récursivement dans un répertoire
-s on|off Active/désactive les rapports à syslog. Défaut : off
-t Spécifie un filtre (regex) à comparer avec le nom de fichier ou du répertoire.
-v mode verbeux
-x Spécifie un fichier ou répertoire exclus
-X Similaire mais en utilisant une expression régulière

Chaînes pour la commande

En utilisant l'option -c, ces chaînes sont disponible :

%c Event cookie number

%e Nom de l'événement

%f Chemin complet du fichier

%F L'ancien nom du fichier (moved_to)

%p Nom du programme (iWatch)

%v Numéro de version

Événements

Les événements suivant sont possibles pour l'options -e :

access Le fichier est accédé

attrib Les attributs sont changés

close Le fichier est fermé

close_nowrite fichier fermé après avoir été ouvert en mode lecture-seule

close_write fichier fermé après avoir été ouvert en mode lecture écriture

create Fichier créé dans le répertoire

delete Fichier supprimé dans le répertoire

delete_self Le fichier supervisé a été supprimé

ignored Le fichier a été ignoré

isdir en événement s'est produit avec le repertoire

modify Le fichier a été modifié

move un fichier/répertoire dans le répertoire recherché a été déplacé

moved_from Un fichier a été déplacé depuis

moved_to Un fichier a été déplacé vers

oneshot Seulement avoyé une fois

open Le fichier a été ouvert

q_overflow La file d'événement déborde

unmount Le système de fichier sur lequel le fichier existe a été démonté

default = close_write, create, delete, move, delete_self et move_self

all_events Tous les événements

Exemples

Monitor les changements dans le /tmp avec les événements pas défaut :
iwatch /tmp
Monitor seulement les événements access et create dans /etc, récursivement, à l'exception de /etc/mail, et envoie un mail à root@example.com
iwatch -r -e access,create -m root@example.com -x /etc/mail /etc
Monitor /bin récursivement, et exécute les commandes w et ps -ef
iwatch -r -c (w;ps -ef)|mail -s '%f was changed' root@localhost /bin
Monitor ~/projects, excluant les répertoires .svn.
iwatch -r -X '.svn' ~/projects
Exemple de fichier de configuration

```
<?xml version="1.0" ?>
 <!DOCTYPE config SYSTEM "/etc/iwatch/iwatch.dtd" >

<config>
```

```
 <guard email="root@example.com" name="iWatch"/>
 <watchlist>
 <title>WEB server integrity monitoring</title>
 <contactpoint email="someone@example.com" name="Administrator"/>
  <path type="recursive" syslog="on" alert="off" exec="echo %p: %e %f | /usr/bin/sendxmpp -t
foo@jabber-br.org">/var/www</path>
  <path type="exception">/var/www/counter</path>
 </watchlist>
</config>
```

Les 2 premières lignes définissent la version XML et le fichier qui définis le motif utilisé par iWatch (défaut : /etc/iwatch/iwatch.dtd).

La déclaration <config> est utilisé pour marquer le port de départ de la configuration. La ligne guard email est utilisée pour spécifier l'email et le nom du champ From :. watchlist délimite un block de définitions à superviser. La déclaration watchlist peut être spécifié plusieurs fois.

title est utilisé pour ajouter un titre pour identifier le block. contactpoint est l'email du contact. path peut monitorer un fichier/répertoire ou exécuter des actions.

Autre exemple possédant 3 watchlist :

```
<?xml version="1.0" ?>
<!DOCTYPE config SYSTEM "iwatch.dtd">

<config>
 <guard email="admin@localhost" name="iWatch"></guard>
 <watchlist>
  <title>Public Website</title>
  <contactpoint email="webmaster@example.com" name="WebMaster"/>
  <path type="single">/var/www/localhost/htdocs</path>
  <path type="single" syslog="on">/var/www/localhost/htdocs/About</path>
  <path type="recursive">/var/www/localhost/htdocs/Photos</path>
 </watchlist>
 <watchlist>
  <title>Operating System</title>
  <contactpoint email="admin@localhost" name="Administrator"/>
  <path type="recursive">/etc/apache2</path>
  <path type="single">/etc/passwd</path>
  <path type="recursive">/etc/mail</path>
  <path type="exception">/etc/mail/statistics</path>
  <path type="single" filter="shadow|passwd">/etc</path>
 </watchlist>
 <watchlist>
  <title>Only Test</title>
  <contactpoint email="root@localhost" name="Administrator"/>
  <path type="single" alert="off" exec="(w;ps -ef)|mail -s %f root@localhost">/tmp/dir1</path>
  <path type="single" events="access,close" alert="off" exec="(w;ps -ef)|mail -s %f
root@localhost">/tmp/dir2</path>
  <path type="single" events="default,access" alert="off" exec="(w;ps -ef)|mail -s '%f is accessed'
root@localhost">/tmp/dir3</path>
  <path type="single" events="all_events" alert="off">/tmp/dir4</path>
 </watchlist>
</config>
```