DESCRIPTION DE LA DEMANDE :
L'analyse des observateurs d'événements est une tâche fastidieuse et très longue.
Cet article a pour but de vous proposer une solution permettant :
- De collecter tous une ou partie des observateurs d'événements de vos serveurs
- D'exporter les logs sous forme d'un fichier CSV.

1. PRESENTATION DE LA FONCTIONNALITE DE SOUSCRIPTION :

Depuis Windows Vista / Windows 2008, l'observateur d'événements dispose d'une fonctionnalité appelée « *Souscription* ».
Cette fonctionnalité permet de récupérer le contenu des observateurs d'événements d'une ou plusieurs machines sous Windows XP Pro, Windows Vista, Windows 2003, Windows 2008, Windows 2008 R2.
Il est possible de créer des filtres pour ne récupérer que certains observateurs d'événements (protocole XQUERY).

2. PRESENTATION DE LA SOLUTION POUR EXPORTER LES OBSERVATEURS D'EVENEMENTS AU FORMAT CSV :

Les commandes PowerShell *Get-Eventlog* et *Get-WinEvent* permettent de lister le contenu des journaux d'évènements.
Nous verrons dans cet article qu'il est nécessaire d'utiliser la commande PowerShell *Get-WinEvent* pour analyser le journal « *Forwarded Events* » (journal utilisé pour collecter les événements).

3. MISE EN PLACE DE LA SOUSCRIPTION :

Le serveur collecteur et les serveurs « sources » doivent être dans le même domaine ou deux domaines différents mais qui s'approuvent.

SUR LE SERVEUR DE SOUSCRIPTION :
Taper les commandes suivantes :
winrm quickconfig
wecutil qc
Ajouter l'entité de sécurité « *NETWORK SERVICE* » dans le groupe « *Event Log Readers* ».

SUR LES TOUS LES ORDINATEURS « SOURCES » :
Taper la commande suivante :
winrm quickconfig
Ajouter l'entité de sécurité « *NETWORK SERVICE* » dans le groupe « *Event Log Readers* ».
Ajouter le compte ordinateur du serveur collecteur en tant qu'administrateur local de la machine source (membre du groupe « Administrators »).

Créer et configurer la souscription sur le collecteur :
Démarrer la console « *Server Manager* ».
Aller dans l'onglet « *Diagnostics\Event Viewer\Subscription* ».
Faire un clic droit et sélectionner « *Create Subscription* ».
Sélectionner « *Forwarded Events* » comme « *Destination log* ».
Sélectionner « *Collector initiated* »et cliquer ensuite sur « Select Computers ».
Ajouter tous les machines pour lesquels vous voulez collecter les logs.
Cliquer sur « *Add Domain computers* ».
Entrer le nom de vos serveurs.
Cliquer sur le bouton « *Test* » pour vérifier que la source est accessible depuis le collecteur.
Au niveau des propriétés de la souscription, cliquer sur le bouton « *Select Events* ».
Au niveau de « *QUERY* », créer votre requête pour récupérer que les logs.
Aller ensuite dans l'onglet « *XML* »
Cocher la case « Edit query manually ».Il est en possible d'affiner les requêtes par défaut (via l'interface) pour collecter les événements.
Simplifier la requête *XQUERY*.
Faire OK pour valider.

Configurer le collecteur pour recevoir les événements au format binaire.
Taper les commandes suivantes :
wecutil es : permet de déterminer le nom de la souscription
wecutil gs ars : permet de voir les paramètres de la souscription
wecutil ss ARS /cf:events : permet de configurer la souscription pour récupérer les événements au format binaire.

Redémarrer ensuite les serveurs sources et le collecteur (si les descriptions n'apparaissent pas correctement).
Vérifier que la souscription fonctionne. Pour cela, faire un clic droit sur la souscription et aller dans « *RunTime Status* ».

4. SCRIPT POUR ANALYSER LES LOGS :

Le script suivant permet d'extraire sous forme d'un fichier CSV tous les événements des dernières 24 heures.
Il permet d'afficher la description de l'événement sur une seule ligne (nécessaire pour créer le fichier CSV).
Il crée un fichier résultat avec la date d'hier.

```
# Date Intialization with format YYYY-MM-DD
$Yesterday = (get-date (get-date).AddDays(-1) -format yyyy-MM-dd)

# Create Log file of the day
$file = New-Item -type File ("C:\MSREPORT\" + $YESTERDAY +"_LOGS.csv")

# Initialize header line in CSV file
$HeaderLine = "DateandTime;Source;EventID;TaskCategory;Message" | Out-File $file -Append

$filterXml = '
  <QueryList>
  <Query Id="0" Path="ForwardedEvents">
    <Select Path="ForwardedEvents">*[System[TimeCreated[timediff(@SystemTime) &lt;=
2592000000]]]</Select>
  </Query>
</QueryList>'

Get-WinEvent –FilterXml $filterXml | select
TimeCreated,ProviderName,Id,TaskDisplayName,Message | foreach {
        $Message = $_.Message -replace('\t'," ")
        $Message = $Message -replace ("\s" , " ")
        $Message = $Message.replace("`r`n"," ")
        $LineToWrite = [string]$_.TimeCreated + ";" + $_.ProviderName + ";" + [string]$_.Id + ";" +
$_.TaskDisplayName + ";" + $Message | Out-File $file -Append
}
```

5. LES RETOURS D'EXPERIENCES :
La fonctionnalité de souscription est plus complexe à mettre en place qu'il n'y paraît !

RETOUR D'EXPERIENCE 1 : LA COLLECTE DES JOURNAUX D'EVENEMENTS FONCTIONNE MAIS LA DESCRIPTION DE CHAQUE EVENEMENT EST INCORRECTE.

On arrive dans ce cas à collecter les événements mais le message dans le champ description est remplacé par **«** *The description for Event ID XXXX from source XXXX cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer »*

Log Name: System
Source: Service Control Manager
Date: 16/08/2011 17:11:23
Event ID: 7036

Task Category: None
Level: Information
Keywords: Classic
User: N/A
Computer: FR92SV0001.newlife.lan
Description:
The description for Event ID 7036 from source Service Control Manager cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.
If the event originated on another computer, the display information had to be saved with the event.
The following information was included with the event:
WinHTTP Web Proxy Auto-Discovery Service

POUR CORRIGER CE PROBLEME :

Effectuer les actions suivantes sur le serveur qui collecte les événements et sur toutes les machines sources :
Ajouter l'entité de sécurité « *NETWORK SERVICE*» (service réseau) dans le groupe « *Event Log Readers* **».** On retrouve ce groupe au niveau de la base de compte locale de chaque serveur en groupe de travail / membre d'un domaine et dans le conteneur BUILTIN de l'annuaire Active Directory sur un contrôleur de domaine.

Configurer la souscription pour récupérer les événements au format binaire et non au format texte :
Sur le serveur qui collecte les événements, taper les commandes suivantes :
wecutil es : permet de déterminer le nom de la souscription
wecutil gs ars : permet de voir les paramètres de la souscription
wecutil ss ARS /cf:events : permet de configurer la souscription pour récupérer les événements au format binaire.

Configurer le serveur de collecte pour disposer des descriptions des événements :
Si vous collectez les événements d'un contrôleur de domaine, il sera nécessaire d'installer la console d'administration du rôle « *Active Directory Service* ». Pour faire cela sous Windows 2008, aller dans le « *Server Manager* | *Features* ». Faire un clic droit avec la souris et sélectionner « *Add features* ». Développer « *Remote Server Administration Tools* » et sélectionner les consoles d'administration à installer.

RETOUR D'EXPERIENCE 2 : LE MESSAGE D'ERREUR ACCESS DENIED APPARAIT :
Dans certains cas la souscription ne fonctionne pas et on a le message « *Access is denied* ».
Cela se produit si vous configurer la souscription sur « *Minimize latency* » au niveau des paramètres avancées de la souscription.
Pour plus d'informations :
http://social.technet.microsoft.com/Forums/en-US/winserverManagement/thread/665375cc-5cb6-4b5b-952e-1c9908382208

RETOUR D'EXPERIENCE 3 : IL N'EST PAS POSSIBLE DE COLLECTER LES LOGS VERS UN JOURNAL PERSONNALISE :
Il est possible de créer un observateur d'événement personnalisé mais il ne semble pas être possible de collecter les logs vers ce journal.
La commande suivante permet par exemple de créer un journal personnalisé appelé MSREPORT : *new-eventlog -source MSREPORT -logname MSREPORT*

Dans cet exemple, les événements seront collectés dans le journal « *FORWARDED EVENTS* ».

Pour plus d'informations :
http://social.technet.microsoft.com/Forums/en-US/winserverManagement/thread/f16be533-4f4a-469e-bc17-7591eb46461b/

RETOUR D'EXPERIENCE 4 : IMPOSSIBLE D'ANALYSER LE JOURNAL « FORWARDED EVENTS » AVEC LA COMMANDE POWERSHELL GET-EVENTLOG :

Depuis Windows 2008, il y a deux types de journaux d'événements : « *SYSTEM* » et « *OPERATIONAL* ».

Hors la commande PowerShell *Get-EventLog* ne permet que d'analyser les journaux d'événements de type « SYSTEM ». Il n'est donc pas possible d'analyser le journal « *FORWARDED EVENTS* » avec cette commande.
Il faut pour cela utiliser la commande PowerShell Get-WinEvent.
Le paramètre -FilterHashtable permet d'effectuer les mêmes filtres que la commande Get-EventLog.

RETOUR D'EXPERIENCE 5 : LE PARAMETRE « –FILTERHASTABLE START TIME » NE MARCHE PAS AVEC LE JOURNAL « FORWARDED EVENTS » :

Lancer PowerShell et exécuter les deux commandes ci-dessous :
$starttime = (get-date).adddays(-1)
$endtime = (get-date)

Exécuter maintenant la commande suivante :
Get-WinEvent -FilterHashtable @{logname="application"; starttime=$starttime ; EndTime= $endtime}
Vous obtenez alors tous les événements contenus dans le journal application des dernières 24 heures.
Si votre journal d'événement est vide ou que vous n'avez aucun événement depuis 24 heures, vous obtenez le résultat suivant :
Get-WinEvent : No events were found that match the specified selection criteria

Assurez-vous maintenant que vous disposez d'au moins 1 événement depuis 24 heures dans le journal « *Forwarded Events* » et lancer la commande PowerShell suivante :
Get-WinEvent -FilterHashtable @{logname="ForwardedEvents"; starttime=$starttime ; EndTime= $endtime}
Cela renvoie systématiquement « *Get-WinEvent : No events were found that match the specified selection criteria ».*
Le problème vient en fait du fait que le journal *Forwarded Events* rencontre un problème avec le paramètre *Start Time*.
On peut reproduire le problème directement dans l'interface graphique de Windows Server 2008.
Pour cela :
Lancer la console « *Event Viewer* ». Faire un clic droit sur « *Forwarded Events* » et sélectionner « *Filter Current Log* ». Dans la liste déroulante « *Logged* », sélectionner « *Custom Range* ».
Définir la date de démarrage (FROM) sur « Events on » et spécifier une date.
Définir la date de fin (TO) sur « Last Events ».
Faire OK. Aucun évènement ne s'affiche.

Pour contourner ce problème, il faut sélectionner dans le filtre « Last 24 hours ».
Si vous voulez une période de temps personnalisé, au niveau de votre filtre cliquer sur l'onglet XML et copier / coller le contenu du filtre (*QueryList*).

```
<QueryList>
  <Query Id="0" Path="ForwardedEvents">
    <Select Path="ForwardedEvents">*[System[TimeCreated[timediff(@SystemTime) &lt;= 86400000]]]</Select>
  </Query>
</QueryList>
```

La commande PowerShell Get-WinEvent peut être utilisée avec un filtre personnalisé :

```
$filterXml = '
   <QueryList>
  <Query Id="0" Path="ForwardedEvents">
    <Select Path="ForwardedEvents">*[System[TimeCreated[timediff(@SystemTime) &lt;=
2592000000]]]</Select>
  </Query>
</QueryList>'
Get-WinEvent –FilterXml $filterXml
```

Cela marche maintenant !

Pour plus d'informations :

get-Help Get-Winevent -full > c:\getwinvent.txt
Lire ensuite le fichier *c:\getwinvent.txt*
http://blogs.msdn.com/b/powershell/archive/2011/04/14/using-get-winevent-filterxml-to-process-windows-events.aspx
http://stackoverflow.com/questions/9224527/get-time-of-forwarded-events-with-get-winevent
http://msdn.microsoft.com/en-us/library/aa385231.aspx

RETOUR D'EXPERIENCE 6 : IL N'EST PAS POSSIBLE DE SUPPRIMER JUSTE UN EVENEMENT DANS UN JOURNAL :

Il ne semble pas être possible de supprimer uniquement un seul événement pour des raisons de sécurité. En effet les commandes *Clear-EventLog* et *Remove-EventLog* permettent uniquement de supprimer tous les événements d'un journal de logs.

RETOUR D'EXPERIENCE 7 :
La souscription échoue. On peut voir le message d'erreur ci-dessous en faisant un clic droit sur la souscription puis « *RunTime Status* »
[FQDN] - Error - Last retry time: 02/06/2012 13:23:15. Code (0x138C): <f:ProviderFault provider="Event Forwarding Plugin" path="%systemroot%\system32\wevtfwd.dll" xmlns:f="http://schemas.microsoft.com/wbem/wsman/1/wsmanfault"><t:ProviderError xmlns:t="http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog">Windows Event Forward plugin can't read any event from the query since the query returns no active channel. Please check channels in the query and make sure they exist and you have access to them.</t:ProviderError></f:ProviderFault> Next retry time: 02/06/2012 13:28:15.

Le problème provient en fait du fait que l'on a fait une requête XML QUERY avec un filtre sur le champ Category.
Dans mon cas la requête échoue quand je sélectionne plus de 22 critères. Je pense qu'il s'agit d'un problème de longueur de requête. Pour contourner le problème j'utilise les opérateurs **=** ou != selon le cas pour réduire la taille de la requête.

```
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='ESENT'] and (Level=1  or Level=2 or Level=3 or Level=4 or Level=0 or Level=5) and ( Task = 1 or Task = 2 or Task = 3 or Task = 4 or Task = 5 or Task = 6 or Task = 7 or Task = 8 or Task = 9 or Task = 10 or Task = 12 or Task = 13 or Task = 14 or Task = 15 or Task = 16 or Task = 17 or Task = 18 or Task = 19 or Task = 20 or Task = 21 or Task = 23 or Task = 24 or Task = 25 or Task = 26 or Task = 27 or Task = 28 or Task = 29)]]</Select>
  </Query>
</QueryList>
```

Si je veux toutes les category sauf la catégorie 11, je crée ma requête de cette façon et cela marche alors :

```
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='ESENT'] and (Level=1  or Level=2 or Level=3 or Level=4 or Level=0 or Level=5) and ( Task != 11)]]</Select>
  </Query>
</QueryList>
```