Pack Windows

Contexte

Lorsque vous installez Shinken enterprise, un certain nombre de modèles et de commandes sont inclus dans votre configuration.

Le pack "Windows", comme son nom l'indique, permet de superviser des hôtes sur lesquels est installé le système d'exploitation Windows (serveur ou client).

Il contient 15 commandes, 1 modèle de check, et 9 modèles de checks dédiés à 1 modèle d'hôte spécifique (nommé "windows").

Toutes les commandes de ce pack se basent sur le script perl **check_wmi_plus.pl** présent dans le répertoire des scripts shinken **/var/lib/shinken/libexec** (ou **$PLUGINSDIR$** depuis l'interface de configuration).

WMI (Windows Management Instrumentation) est un système de gestion interne de Windows qui prend en charge la surveillance et le contrôle de ressources systèmes via un ensemble d'interfaces. Le script perl permet d'interroger ces interfaces via un nom d'utilisateur et un mot de passe. Si l'utilisateur a les droits suffisants, alors le système d'exploitation Windows retournera l'information demandée.

Nous allons ici détailler ces checks et commandes WMI associés au modèle Windows de ce pack.

Sommaire

Le modèle d'hôte Windows et ses données héritées

Le modèle d'hôte Windows, sur lequel est accroché les différents checks dédiés, contient des données (locales) qui seront utilisés par les checks. Ces données seront invoquées par les checks et commandes via **$_HOST** suivi du nom de la variable.

Exemple : **$_HOSTWINDOWS_ALL_CPU_CRIT$** utilisera la donnée nommée **WINDOWS_ALL_CPU_CRIT** (quelle soit locale ou héritée d'un modèle).

Pour un hôte qui hérite par exemple du modèle windows de notre pack, ces données seront donc héritées également, mais elles pourront aussi être surchargées directement sur l'hôte (attention aux conflits de nom des données).

Si vous souhaitez modifier de manière globale ces données, ou en rajouter, faîtes le directement sur le modèle windows.

Détails des checks et commandes dédiés au modèle Windows

Cpu

Le modèle de check Cpu, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_overall_cpu** :

```
 $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkcpu -w "$_HOSTWINDOWS_ALL_CPU_WARN$" -c "$_HOSTWINDOWS_ALL_CPU_CRIT$" --inidir=$WMI_INI_DIR$
```

Certains scripts qui vérifient les performances CPU via WMI ou SNMP , ne prennent juste que des valeurs précalculées. Ce n'est pas le cas ici. Nous utilisons les compteurs brut WMI pour calculer les valeurs sur une période de temps donnée. C'est bien plus précis que prendre les valeurs WMI formattées.

Exemple de commande avec données interprétées, qui peut être lancée directement depuis le terminal du serveur du démon Poller :

```
/var/lib/shinken/libexec/check_wmi_plus.pl -H "ipdemonserveur" -u "monuser" -p "monpassword" -m checkcpu -w "80" -c "90" --inidir=/var/lib/shinken/libexec/check_wmi_plus.d
```

Note : ici, la commande retournera un OK si l'utilisation CPU est inférieure à 80%, WARNING si l'utilisation CPU est entre 80% et 90%, et CRITIQUE si l'utilisation est supérieure à 90%

Le Check Cpu nécessite la valeur précédente collectée afin de déterminer le statut à retourner. Donc ne vous inquiétez pas si votre Check affiche un message à ce sujet à la première vérification.

Disks

Le modèle de check Disks, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_disks** :

```
 $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkdrivesize -a "."  -w "$_HOSTWINDOWS_DISK_WARN$" -c "$_HOSTWINDOWS_DISK_CRIT$" -o 0 -3 1  --inidir=$PLUGINSDIR$
```

- Le premier argument **-a** permet de choisir une lettre de disque ou un nom de volume pour la vérification. Si non renseigné, une liste des disques valides seront affichés. Si renseigné avec "." tous les disques seront inclus.Pour inclure plusieurs disques, séparez les avec le caractère | (le pipe). Cela fonctionne avec une expression régulière, alors spécifiez précisément ce que vous voulez. Par exemple "C" ou "C:" ou "C|E" ou "." ou "Data"
- Le deuxième argument **-o**

ARG2 Set this to 1 to use volumes names (if they are defined) in plugin output and performance data ie -o 1
ARG3 Set this to 1 to include information about the sum of all disk space on the entire system.
If you set this you can also check warn/crit against the overall disk space.
To show only the overall disk, set ARG3 to 1 and set ARG1 to 1 (actually to any non-existant disk)
Eg -o 1 -3 1

WARN/CRIT can be used as described below.
Valid Warning/Critical Fields are: _Used% (Default), _UsedGB, _Free%, _FreeGB.

EventLogApplication

Le modèle de check EventLogApplication, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_eventlogs** avec un Argument "application" :

```
 $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkeventlog -a "$ARG1$" -o 2 -3 1  -w "$_HOSTWINDOWS_EVENT_LOG_WARN$" -c "$_HOSTWINDOWS_EVENT_LOG_CRIT$" --inidir=$WMI_INI_DIR$
```

*Note : **$ARG1$** sera donc remplacé par **application** lors de l'exécution de la commande*

EventLogSystem

Le modèle de check EventLogSystem, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_eventlogs** avec un Argument "system" :

```
 $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkeventlog -a "$ARG1$" -o 2 -3 1  -w "$_HOSTWINDOWS_EVENT_LOG_WARN$" -c "$_HOSTWINDOWS_EVENT_LOG_CRIT$" --inidir=$WMI_INI_DIR$
```

*Note : **$ARG1$** sera donc remplacé par **system** lors de l'exécution de la commande*

Memory

Le modèle de check Memory, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_physical_memory** :

```
 $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkmem -w "$_HOSTWINDOWS_MEM_WARN$" -c "$_HOSTWINDOWS_MEM_CRIT$" --inidir=$WMI_INI_DIR$
```

Network Interface

Le modèle de check Network Interface, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_network** :

```
 $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checknetwork  -a "$_HOSTWINDOWS_NETWORK_INTERFACE$" --inidir=$PLUGINSDIR$
```

Reboot

Le modèle de check Reboot, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_reboot** :

```
$PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkuptime -w "$_HOSTWINDOWS_REBOOT_WARN$" -c "$_HOSTWINDOWS_REBOOT_CRIT$" --inidir=$WMI_INI_DIR$
```

Services

Le modèle de check Services, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_auto_services** :

```
$PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkservice -a Auto -o "$_HOSTWINDOWS_EXCLUDED_AUTO_SERVICES$" -w "$_HOSTWINDOWS_AUTO_SERVICES_WARN$" -c "$_HOSTWINDOWS_AUTO_SERVICES_CRIT$" --inidir=$WMI_INI_DIR$
```

Voici un exemple de commande qui va vérifier un service spécifique, ici la bonne activité du service de "Pare feu" Windows :

```
$PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkservice -a MpsSvc -c 0 --inidir=$WMI_INI_DIR$
```

Swap

Le modèle de check Swap, dédié au modèle Windows (propriété "Attaché sur les modèles d'hôte" mise à "windows"), utilise la commande **check_windows_swap** :

```
$PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m
checkpage -a auto --inidir=$WMI_INI_DIR$
```

Personnalisation et autres commandes basées sur le script WMI PLUS

Il est possible de personnaliser les commandes ci dessus pour correspondre au mieux à vos besoins.

Disks

Personnalisation de la commande pour aller pointer vers un disque particulier, ici le disque C :

```
$PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m
checkdrivesize -a "C" -w "$_HOSTWINDOWS_DISK_WARN$" -c "$_HOSTWINDOWS_DISK_CRIT$" -o 0 -3 0 --
inidir=$PLUGINSDIR$
```

Vérification de la taille d'un fichier :

Le script suivant va permettre de vérifier la taille du fichier pagefile.sys et retourner un Warning si sa taille dépasse 1500 mo et un Critical si sa taille dépasse 2 go.

En command line shell : ./check_wmi_plus.pl -H "192.168.1.241" -m checkfilesize -u "administrateur" -p "pass" -m checkfilesize -a c:/pagefile.sys -w 1500m -c 2g

Shinken Commands :

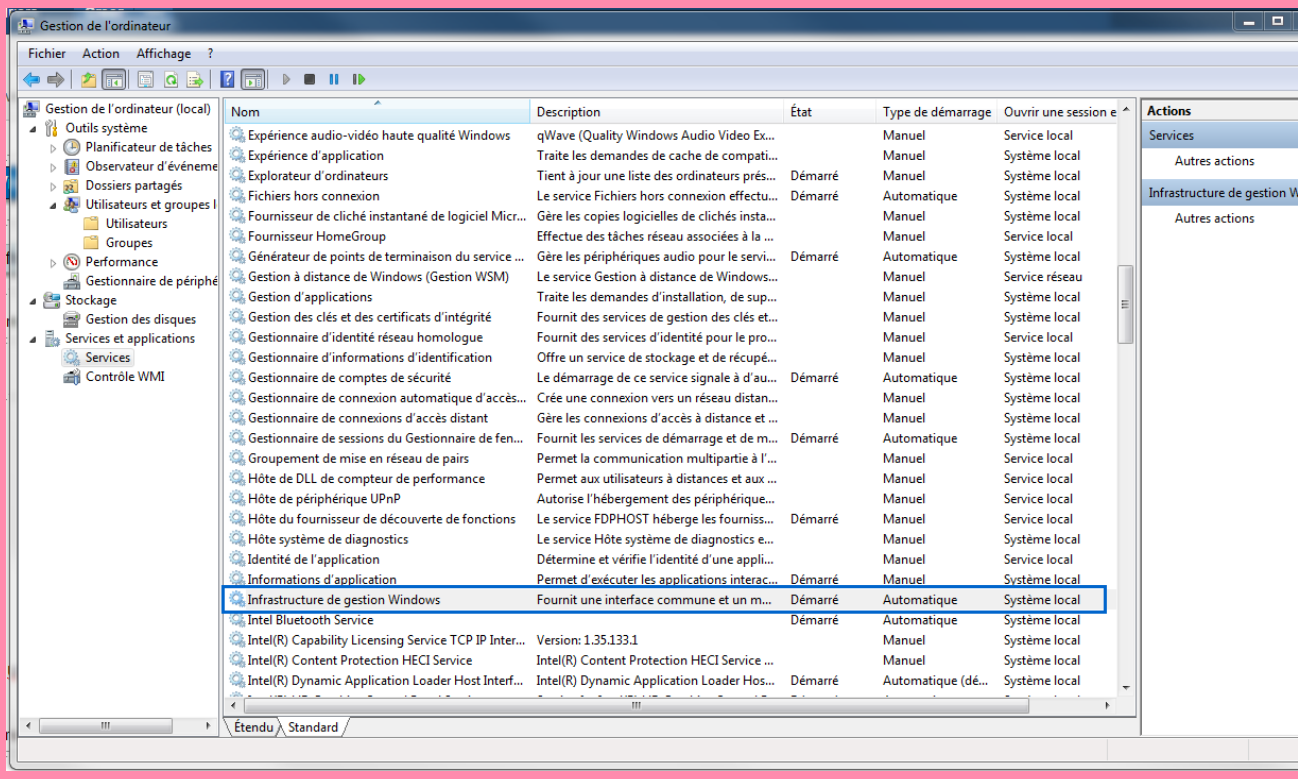
```
via args : $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -m checkfilesize -u "$_HOSTDOMAINUSER$" -p
"$_HOSTDOMAINPASSWORD$" -a $ARG1$ -w $ARG2$ -c $ARG3$  --inidir=$WMI_INI_DIR$

ou

via données d'hôte : $PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -m checkfilesize -u "$_HOSTDOMAINUSER$" -
p "$_HOSTDOMAINPASSWORD$" -a "$_HOSTWINDOWS_FILESIZE_PATH$" -w "$_HOSTWINDOWS_FILESIZE_WARN$" -c
"$_HOSTWINDOWS_FILESIZE_CRIT$" --inidir=$WMI_INI_DIR$
```

Fonctionnement WMI sur poste client ou serveur Windows

Windows Management Instrumentation (infrastructure de gestion windows)

Le service WMI est installé et démarré par défaut sur les systèmes d'exploitations windows.

Vous pouvez aller vérifier si ce service est bien démarré en vous rendant dans le gestionnaire de service windows :



Si vous utilisez des firewall :

- le Poller doit être autorisé à communiquer avec l'hôte supervisé
- les ports WMI de cet hôte doivent être ouverts : les ports TCP 135 et 445 ainsi que des ports dynamiques, typiquement dans le range de 1024 à 1034, doivent être accessibles

WMI Avancé - gestion de la sécurité

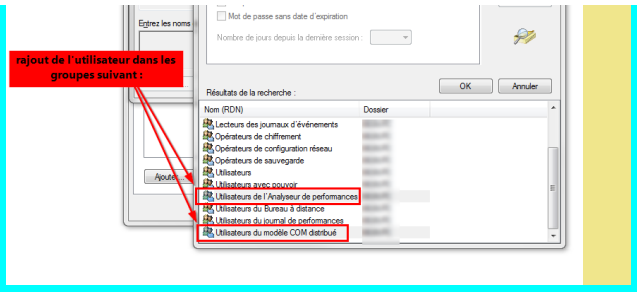
Comme on l'a vu précédemment, les commandes WMI requiert une authentification au préalable afin de récupérer des informations de supervision sur l'hôte Windows. (-u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$")

L'utilisation du compte d'administrateur du poste Windows permet facilement d'obtenir ces informations avec succès car ce compte à tous les droits d'accès (WMI, DCOM etc..).

Cependant, pour des raisons de sécurité, il se peut que vous préfériez utiliser un compte avec des droits plus restreints.

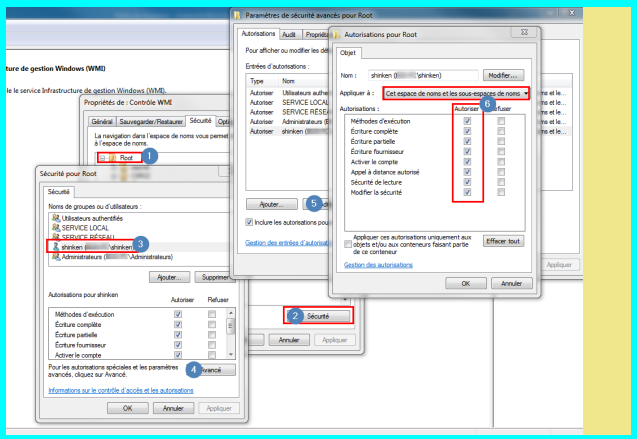
Voici donc la procédure à suivre pour rajouter des droits à un utilisateur basique qui vous servira à récupérer les informations WMI souhaitées.

Une fois que vous avez créé votre utilisateur sur le poste client ou sur votre domaine, ouvrez la console de "gestion de l'ordinateur" (compmgmt.msc) sur le poste à superviser, et rajoutez à l'utilisateur les droits suivants : Utilisateurs du modèle COM distribué (Distributed COM Users) et Utilisateurs de l'Analyseur de performance (Performance Monitor Users) :



Il faut à présent rajouter les droits sur le contrôle WMI, pour cela, depuis la console de "gestion de l'ordinateur":

Cliquez sur Services & Applications
Cliquez sur Contrôle WMI
Clic droit - Propriété
Sélectionnez l'onglet Sécurité
Sélectionnez Root (1)
Cliquez sur le bouton de Sécurité en bas (2)
Rajoutez votre utilisateur (3)
Allez dans les propriétés avancées (4)
Modifiez la sécurité de l'utilisateur (5)
Rajoutez lui toutes les autorisations et l'application doit se faire à "Cet espace de noms et les sous-espaces de noms" (6)
Cliquez sur OK sur toutes les fenêtres.



Veuillez ouvrir la console des services (services.msc) et redémarrez le service gérant la partie WMI : "Infrastructure de gestion Windows".

Votre check (via l'utilisateur spécifique passé en paramètre) doit maintenant pouvoir passer une requête WMI à cet ordinateur depuis votre commande Shinken.

Sur les neuf, un seul check peut poser problème, c'est la requête qui interroge les services Windows via le check "Services". Ce check va vous renvoyer une erreur :
UNKNOWN - The WMI query had problems. The error text from wmic is: [wmi/wmic.c:212:main()] ERROR: Retrieve result data. NTSTATUS: NT code 0x80041003 - NT code 0x80041003

Voyons comment résoudre ce problème dans la prochaine section.

Déléguer des droits d'accès minimum d'un utilisateur sur les services Windows

Pour que l'utilisateur puisse avoir les droits de questionner les différents services Windows, il faut au préalable l'autoriser avec des droits de bases.

Par exemple, il peut être pratique de lister et contrôler uniquement des services clés Windows. Par défaut, les utilisateurs locaux et les comptes non-administrateurs ne possèdent même pas les droits d'agréger les services locaux et encore moins de questionner leurs statuts ou de les redémarrer. Heureusement, il y a un contournement. Il est à noter que cette méthode s'applique pour un compte d'utilisateur, et non pour les groupes de sécurités.

Vous aurez besoin d'un outils spécial appelé "Subinacl" afin de rajouter des permissions, vous pouvez télécharger une copie ici : Microsoft Link. Attention, il vous faudra un compte Administrateur pour procéder aux étapes suivantes.

La première chose à faire est de déterminer le SID (identifiant de sécurité) du compte auquel vous souhaitez ajouter les autorisations. Cela peut être fait de différentes manières, la plus simple étant l'exécution d'une commande wmic :

```
wmic useraccount where name='utilisateur' get sid
```

Une fois que vous avez récupéré le SID du compte sur lequel vous souhaitez rajouter les droits, voici les différentes commandes à exécuter:

1. Ouvrez une ligne de commande en administrateur et exécuter:

```
sc sdshow scmanager
```

1. Copiez l'output (SDDL) et collez le dans un éditeur de texte, ça devrait ressembler à la ligne suivante :

D:(A;;CC;;;AU)**(A;;CCLCRPRC;;;IU)**(A;;CCLCRPRC;;;SU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)S:(AU;FA;KA;;;WD)(AU;OIIOFA;GA;;;WD)

1. Copiez la section entre parenthèses du SDDL incluant le IU (interactive users) et collez là juste avant le S:
2. Remplacez le 'IU' avec le SID de l'utilisateur, ça devrait ressembler à cela :

D:(A;;CC;;;AU)**(A;;CCLCRPRC;;;IU)**(A;;CCLCRPRC;;;SU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)**(A;;CCLCRPRC;;;S-1-5-21-214A909598-1293495619-13Z157935-75714)**S:(AU;FA;KA;;;WD)(AU;OIIOFA;GA;;;WD)

1. Exécutez la commande suivante pour autoriser l'utilisateur du SID spécifié à **énumérer les services locaux Windows** :

```
sc sdset scmanager "D:(A;;CC;;;AU)(A;;CCLCRPRC;;;IU)(A;;CCLCRPRC;;;SU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)(A;;CCLCRPRC;;;S-1-5-21-214A909598-1293495619-13Z157935-75714)S:(AU;FA;KA;;;WD)(AU;OIIOFA;GA;;;WD)"
```

Vous aurez besoin de connaitre le “nom court” du service windows sur lequel vous voulez rajouter les permissions, pour obtenir cela depuis le nom complet du service, via la commande suivante :

```
sc getkeyname "<Nom complet du Service>"
```

Vous pouvez aussi récupérer le nom depuis la MMC de services depuis le panneau de configuration –> Outils d'administration.
Depuis les propriétés du service souhaité, le "Nom du service" correspond au nom court.

Par la suite, l'outil subinacl (que vous avez précédemment téléchargé et installé) va permettre de rajouter les permissions de votre compte d'utilisateur sur le service :

```
subinacl /verbose /service <nom court du service> /grant=<DOMAINE ou MACHINE>\<compte d'utilisateur>=F
```

exemple de rajout des droits sur le service "Pare-feu Windows" pour l'utilisateur MON-PC\USER :

```
subinacl /verbose /service MpsSvc /grant=MON-PC\USER=F
```

Le “=F” permet d'accorder des permissions totales.

Contrairement à l'utilisation du compte administrateur pour la vérification des services (qui permet une vue exhaustive et totale), le rajout des droits d'énumérer les services Windows pour l'utilisateur spécifié ne permet pas de lister et de vérifier la totalité des services.
La commande du pack Windows **check_windows_auto_services** retournera alors l'état des services (en démarrage "automatique") d'un certain nombre de services Windows et de ceux sur lesquels vous avez rajouté les droits de manière manuelle avec la commande subinacl.

Voici un exemple de commande qui va vérifier un service spécifique, ici la bonne activité du service de "Pare feu" Windows :

```
$PLUGINSDIR$/check_wmi_plus.pl -H "$HOSTADDRESS$" -u "$_HOSTDOMAINUSER$" -p "$_HOSTDOMAINPASSWORD$" -m checkservice -a MpsSvc -c 0 --inidir=$WMI_INI_DIR$
```

Résolutions des problèmes

Les checks n'arrivent pas à récupérer les informations alors que l'utilisateur utilisé est Administrateur

Après avoir mis en place le modèle dans Shinken donné les bons droits à l'utilisateur Windows utilisé par le check, il se peut que les données de performances remontées par les checks ne puissent pas être remontées.

Il est possible dans ce cas que les valeurs de la bibliothèque du compteur de performances Windows soit corrompu ou contienne des valeurs incorrectes.

Dans ce cas, les checks Windows peuvent retourner les erreurs suivantes:

- UNKNOWN - The WMI query had problems. The error text from wmic is: [wmi/wmic.c:212:main()] ERROR: Retrieve result data. NTSTATUS: NT code 0x80041017 - NT code 0x80041017
- UNKNOWN - The WMI query had problems. The plugin is having trouble finding the required WMI Classes on the target host (172.16.0.132). There can be multiple reasons for this (please go through them and check) including permissions problems (try using an admin login) or software that creates the class is not installed (eg if you are trying to checkiis but IIS is not installed). It can also happen if your version of Windows does not support this check (this might be because the WMI fields are named differently in different Windows versions). Sometimes, some systems 'lose' WMI Classes and you might need to rebuild your WMI repository. Sometimes the WMI service is not running, other times a reboot car fix it. Other causes include mistyping the WMI namesspace/class/fieldnames. There may be other causes as well. You can use wmic from the command line to troubleshoot. Wmic error text on the next line. [wmi/wmic.c:212:main()] ERROR: Retrieve result data. NTSTATUS: NT code 0x80041010 - NT code 0x80041010

Il est possible de recréer manuellement les valeurs de la bibliothèque du compteur de Performance avec la commande suivante:

```
lodctr /r
```

Plus d'informations sur la commande et ses possibilités peuvent être trouvées sur la page de documentation dédiée: https://docs.microsoft.com/en-u /windows-server/administration/windows-commands/lodctr