L'audit des GPO

Aurélien Bordes
aurelien26@free.fr

Résumé. Les GPO sont apparues avec Windows 2000 et l'Active Directory. Elles permettent d'appliquer des paramètres de configuration sur les ordinateurs et les utilisateurs et d'être distribuées au moyen d'un domaine Active Directory.

Très appréciées et largement utilisées, il n'est pas rare de voir, avec le temps, des domaines définir plusieurs milliers de GPO : il devient alors très difficile de maîtriser tous les paramètres appliqués ou de détecter des problèmes de configuration.

Cet article se propose d'expliquer le fonctionnement des GPO puis de fournir différents points d'audit sur la forme (configuration des GPO) et sur le fond (paramètres définis par les GPO).

Afin d'aider à l'analyse, deux outils sont également décrits et mis à disposition. Le premier, `gpocheck` permet de détecter différents problèmes de configuration, en particulier d'éventuelles désynchronisations de droits entre l'Active Directory et le répertoire SYSVOL. Le second, `gpo2sql` permet d'importer dans une base SQL tous les paramètres définis par un ensemble de GPO afin d'en faciliter leur analyse.

1 Introduction

La gestion d'un parc informatique est une tâche complexe à laquelle sont soumis tous les administrateurs. Cette gestion inclut en particulier la gestion unifiée des ordinateurs ou des profils utilisateur.

Microsoft a toujours incorporé à Windows des mécanismes de gestion. Historiquement basés sur les fichiers `NTCONFIG.POL`, ceux-ci ont été remplacés avec Windows 2000 et l'Active Directory par les GPO (*Group Policy Object* ou stratégies de groupe).

Les GPO sont un mécanisme puissant et extensible permettant de gérer de très nombreux paramètres d'un système Windows (configuration, logiciels installés, etc.), partie appelée « configuration ordinateur », ou d'un profil utilisateur, partie appelée « configuration utilisateur ».

> Note : les recommandations formulées dans l'article sont signalées par un encadré bleu.

2 Fonctionnement des GPO

2.1 Définition des GPO

Dans un domaine Active Directory, les GPO sont définies par des objets de classe `groupPolicyContainer`. Le schéma impose que ce type d'objet soit contenu sous un objet de classe `container` et, dans chaque domaine, le conteneur `CN=Policies,CN=System,<Domain NC[1]>` est dévolu à ce rôle.

Les attributs autorisés par la classe `groupPolicyContainer` permettent de définir les propriétés d'une GPO :

- `cn` : l'identifiant unique de la GPO sous la forme d'un GUID ;
- `displayName` : le nom de la GPO ;
- `nTSecurityDescriptor` : le descripteur de sécurité indiquant en particulier les comptes autorisés à appliquer la GPO ou ceux autorisés à la modifier (voir section 3.1 pour la description des droits) ;
- `versionNumber` : la version (au sens révision) de la GPO composée de la version ordinateur (16 bits de poids faible) et de la version utilisateur (16 bits de poids fort) (voir section 2.9) ;
- `flags` : les flags indiquent si :
  - la partie utilisateur de la GPO est désactivée (valeur `1`),
  - la partie ordinateur de la GPO est désactivée (valeur `2`),
  - la GPO est totalement désactivée (valeur `3` soit `1&2`) ;
- `gPCFunctionalityVersion` : le niveau fonctionnel de la GPO (cet attribut vaut toujours à `2`) ;
- `gPCMachineExtensionNames` : la liste des CSE[2] (voir section 2.5 pour la définition et le rôle des CSE) devant être activés pour l'application ainsi que la liste des objets COM nécessaires pour l'édition de la partie ordinateur de la GPO ;
- `gPCUserExtensionNames` : cet attribut a le même format que `gPCMachineExtensionNames` mais pour l'application ou l'édition de la partie utilisateur de la GPO ;
- `gPCFileSysPath` : le chemin du partage réseau où sont stockés les fichiers de configuration de la GPO (voir section 2.2 pour le rôle et la description du contenu de ce répertoire). Cet emplacement sera appelé « répertoire de la GPO » dans la suite de l'article ;
- `gPCWQLFilter` : la liste des filtres WMI permettant de restreindre l'application de la GPO (voir section 2.7).

1. *Domaine Naming Context*, c'est-à-dire le nom de base de la partition LDAP correspondant à un domaine.
2. *Client Side Extensions.*

Ainsi, pour récupérer dans un domaine toutes les GPO et les propriétés associées, il suffit de requêter, à la racine du *naming context* d'un domaine, tous les objets de type `groupPolicyContainer` via une requête LDAP et avec le filtre `(objectClass=groupPolicyContainer)`. Tous les objets récupérés doivent être situés sous le conteneur `CN=Policies,CN=System`. Les attributs de chaque objet permettent d'obtenir les propriétés de la GPO et de vérifier les différents points d'audit de l'article.

En complément des GPO du domaine, il existe pour chaque système Windows plusieurs GPO locales. Ces GPO permettent d'appliquer des paramètres, y compris pour un système hors domaine ou un compte local. Ces GPO locales sont :

- la « *Local GPO* » (ou LGPO) du système qui permet d'appliquer des paramètres à l'ordinateur et à tous les comptes locaux. Son emplacement disque, permettant de stocker sa configuration et ses paramètres, est située dans `%SystemRoot%\System32\GroupPolicy` ;
- les « *Multiple Local GPO* » (ou MLGPO [11]) qui permettent d'appliquer des paramètres à un groupe ou un utilisateur en particulier. Leur emplacement disque est `%SystemRoot%\System32\GroupPolicyUsers\<SID>` :
  - une MLGPO de groupe pour *Administrators* (`S-1-5-32-544`) ou *Users* (`S-1-5-32-545`)[3]. Une seule des deux MLGPO de groupe sera appliquée suivant l'appartenance ou non au groupe *Administrators*,
  - une MLGPO utilisateur pour chaque compte local.

Les GPO locales ne pouvant pas utiliser l'annuaire pour stocker leur configuration, celle-ci est positionnée dans le fichier `gpt.ini` situé à la racine de l'emplacement disque de chaque GPO locale. On retrouve, dans la section `[General]`, une partie des attributs d'un objet GPO Active Directory : `gPCMachineExtensionNames`, `gPCUserExtensionNames`, `Version` et `Options`.

2.2 Stockage des paramètres des GPO

Si, dans l'annuaire, les objets de type `groupPolicyContainer` définissent les GPO et leurs propriétés associées, il faut également disposer d'espaces de stockage (*Group Policy Storage*) pour stocker les paramètres définis par les GPO. Deux types de stockage peuvent être mis en œuvre.

Le premier est l'annuaire Active Directory. Dans ce cas, les paramètres prennent la forme d'objets avec des classes particulières. Par exemple

3. Uniquement ces deux groupes intégrés peuvent être utilisés.

la classe `ms-net-ieee8023-Policy` pour les *Wired Network Policies*, la classe `ms-net-ieee80211-Policy` pour les *Wireless Network Policies* ou encore la classe `Ipsec-Policy` pour les politiques IPsec. Ce format de stockage présente l'avantage d'être normalisé et facilement requêtable par LDAP. Cependant, ce format n'est pas adapté à de gros volumes ou à des données complexes. Certaines documentations Microsoft référencent ce type de stockage sous le nom *Group Policy container* (GPC). Ces objets sont principalement situés sous l'objet de la GPO.

Note : certains paramètres ne sont pas stockés sous l'objet GPO, c'est-à-dire sous le conteneur `CN=Policies,CN=System`. C'est par exemple le cas des politiques IPsec qui sont stockées sous le conteneur `CN=IP Security,CN=System` [5]. Cette spécificité explique pourquoi certains paramètres ne peuvent pas être modifiés même en ayant le contrôle total sur une GPO, le conteneur `CN=IP Security` n'étant modifiable, par défaut, que par les administrateurs du domaine.

Le deuxième type de stockage est par fichiers : pour chaque GPO, un répertoire est dédié au stockage des fichiers de configuration de la GPO. Comme vu dans la section 2.1, l'attribut `gPCFileSysPath` définit, pour chaque GPO, l'emplacement de ce répertoire. Chaque GPO dispose donc de son propre répertoire. Certaines documentations Microsoft référencent ce type de stockage sous le nom *Group Policy template* (GPT). C'est généralement ce second type de stockage qui est utilisé pour conserver la majorité des paramètres.

Les répertoires des GPO devant être accessibles à tous les ordinateurs et les utilisateurs du domaine, il est systématiquement défini sous la forme `\\<domain.tld>\SYSOL\Policies\<GUID_GPO>`. Pour rappel, le partage `SYSVOL` est un répertoire que chaque contrôleur de domaine partage. La réplication se fait via le protocole DFS-R (*Distributed File System Replication*) et le nom est résolu via DFS-N (*Distributed File System Namespaces*).

Note : NTFRS était utilisé préalablement pour la réplication du SYSVOL. Cependant, ce protocole était jugé « fragile » et a été remplacé par DFS-R avec Windows Server 2008. Sauf à vivre dans obsolescence, DFS-R doit être utilisé pour la réplication du SYSVOL et il faut s'assurer que la migration vers DFS-R a bien été effectuée.

> S'assurer que la réplication du SYSVOL est assurée par DFS-R et que NTFRS n'est plus utilisé.

Pour chaque emplacement de base d'une GPO (dans l'AD et le SYSVOL), il existe toujours deux sous-conteneurs ou sous-répertoires nommés

« Machine » et « User » destinés à recevoir respectivement les paramètres pour la partie ordinateur et la partie utilisateur de la GPO.

Concernant les GPO locales, comme vu dans la section 2.1, celles-ci ne peuvent pas utiliser l'annuaire pour stocker leurs paramètres. Ainsi, tous les paramètres stockés sous forme d'objets dans l'annuaire (*Network Policies*, *Software installation*, *Public Key Policies*, etc.) ne peuvent pas être utilisés et n'apparaissent pas dans l'édition d'une GPO locale. Tous les autres paramètres sont stockés dans l'emplacement disque en reprenant la même structure de fichiers qu'une GPO issue de l'annuaire.

2.3 Liaison des GPO

En complément de la définition des GPO, il faut également définir sur quels ordinateurs ou utilisateurs celles-ci doivent s'appliquer. L'application est paramétrée par les attributs `gPLink` et `gPOptions` qui peuvent être positionnés sur des objets de classe :

- `organizationalUnit` pour appliquer aux ordinateurs ou utilisateurs situés sous une unité d'organisation (OU) ;
- `samDomain` pour appliquer à tous les ordinateurs ou utilisateurs d'un domaine ;
- `site` pour appliquer aux ordinateurs ou utilisateurs d'un site Active Directory.

L'attribut `gPLink`, qui n'est pas multivalué, permet de référencer une ou plusieurs GPO à appliquer avec, pour chaque GPO liée, des options. Le format de l'attribut `gPLink` est le suivant :

```
[<GPO DN_1>;<GPLinkOptions_1>][<GPO DN_2>;<GPLinkOptions_2>]...
[<GPO DN_n>;<GPLinkOptions_n>]
```

`<GPO DN>` référence un objet GPO dans l'annuaire sous forme `LDAP://cn=<GUID_GPO>,cn=policies,cn=system,<domain NC>` et `GPLinkOptions` permet d'indiquer les options de la liaison (cf. tableau 1).

Bit de `GPLinkOptions`	Attributs
1	Le lien de la GPO est désactivé : la GPO n'est pas appliquée.
2	La GPO est en mode « *Enforced* » ce qui permet d'augmenter sa priorité ou d'empêcher son blocage par une unité d'organisation (cf. section 2.13). On utilise ici le terme anglais d'*Enforced* et pas la traduction française « d'appliquée » qui prête énormément à confusion.

Tableau 1. Options de `GPLinkOptions`.

Quant à l'attribut `gPOptions`, il indique les propriétés du conteneur (domaine, unité d'organisation et site) vis-à-vis de l'application des GPO. La seule valeur actuellement définie est `1`, qui indique que le conteneur est « bloquant », ce qui permet de ne pas appliquer certaines GPO (voir section 2.13).

2.4 Moteur GPO client

Côté client, l'application des GPO est mise en œuvre par le service `GPSvc`. Vu la nature des modifications que ce service doit effectuer, il s'exécute dans le contexte `LocalSystem` pour avoir le plus haut niveau de droits et privilèges.

Cependant, dans le cadre de l'application des GPO d'un utilisateur, le service emprunte, via le mécanisme de l'*impersonation*, l'identité de l'utilisateur durant certaines phases. Cela concerne l'accès :

— à l'annuaire pour déterminer les GPO qui doivent s'appliquer ;
— à l'annuaire ou au SYSVOL pour la lecture des paramètres ;
— au Registre pour la lecture ou l'écriture de l'état d'application.

Pour l'application des GPO ordinateur, c'est le contexte d'authentification du compte machine (session d'authentification `0x3e7`) qui est utilisé. Par défaut, ce contexte permet l'authentification au nom du compte de la machine uniquement pour le SSP Kerberos.

Dans tous les cas, la fonction de chaque CSE qui applique les paramètres (`ProcessGroupPolicy(Ex)`) est appelée dans le contexte de sécurité `LocalSystem`. C'est en particulier nécessaire pour modifier les différentes clés de Registre *Policies* des profils utilisateur auxquelles les utilisateurs n'ont qu'un accès en lecture (`HKCU\Software\Policies`, `HKCU\Software\Microsoft\Windows\CurrentVersion\Policies` ou `HKCU\System\CurrentControlSet\Policies`).

Il existe cependant un cas particulier pour le contexte d'exécution du programme `gpscript.exe` en charge de l'exécution des scripts configurés par le CSE `Scripts`. Pour les GPO d'un utilisateur, le moteur GPO lance ce programme avec le contexte de sécurité de l'utilisateur et dans sa session Windows [4] afin que les scripts soient exécutés dans le contexte de l'utilisateur.

2.5 *Client Side Extension*

Les GPO ont été conçues comme un système modulaire et extensible. Ainsi, chaque type de paramètres est appliqué par un composant dédié

4. Ceci est possible grâce au privilège `SeTcbPrivilege`.

à cette tâche appelé CSE (*Client Side Extension*). Les CSE prennent la forme de bibliothèques dynamiques qui sont chargées par le moteur GPO pour appliquer un type de paramètre particulier.

Plus d'une cinquantaine de CSE sont livrés dans une installation par défaut de Windows. Ceux-ci sont identifiés par un GUID et enregistrés dans le Registre sous la clé `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtension`[5]. Pour chaque entrée de CSE dans le Registre, sous la clé correspondant au GUID du CSE, on trouve différentes valeurs caractérisant le CSE (la définition de tous les paramètres est disponible en [4]) :

- valeur par défaut de la clé : nom du CSE ;
- `DisplayName` : nom long d'affichage du CSE ;
- `DllName` : nom de la bibliothèque dynamique qui contient le code d'application du CSE et qui sera chargée dans le moteur GPO ;
- `ProcessGroupPolicy`/`ProcessGroupPolicyEx` : noms des fonctions exportées par la bibliothèque qui seront appelées par le moteur GPO pour appliquer les paramètres du CSE. Ainsi, une même bibliothèque, identifiée par `DllName`, peut contenir plusieurs CSE ;
- `EnableAsynchronousProcessing`/`NoBackgroundPolicy` : ces booléens spécifient la compatibilité du CSE avec les trois modes d'application (voir section 2.6) ;
- `NoGpoListChanges` : ce booléen indique que le CSE ne doit pas être appelé si aucune modification n'est détectée dans les GPO appliquées (voir section 2.9) ;
- `MaxNoGPOListChangesInterval` : indique le temps en minutes au-delà duquel le CSE doit appliquer les GPO même si celles-ci ont déjà été appliquées et qu'aucune modification n'a été détectée (voir section 2.9). Ceci est destiné aux CSE liés à la sécurité pour réappliquer systématiquement les paramètres définis par les GPO. Par défaut, les CSE concernés sont *Security*, *Audit Policy Configuration* et *Central Access Policy Configuration* ;
- `NoMachinePolicy`/`NoUserPolicy` : ces booléens indiquent que le CSE ne gère pas l'application de paramètre ordinateur ou utilisateur afin d'éviter le chargement inutile du CSE lors de l'application des paramètres d'un contexte donné ;
- `PerUserLocalSettings` : si ce booléen est activé, la base d'état du CSE (voir section 2.9) n'est pas située sous la clé HKLM mais sous la clé HKU ;

5. La clé WinLogon est utilisée ici car, jusqu'à Windows Server 2003, c'était WinLogon qui était responsable de l'application des GPO.

— `RequiresSuccessfulRegistry` : ce booléen indique au moteur GPO, lorsqu'il initialise le CSE, de vérifier que le CSE `Registre` (voir section 2.14) s'est correctement exécuté. Pour cela, la valeur `Status`, qui indique le code de retour d'exécution, sous la clé `\Status\GPExtensions\{35378EAC-683F-11D2-A89A-00C04FBBCFA2}` dans la base d'état des GPO (voir section 2.8) doit valoir `ERROR_SUCCESS`. En cas d'erreur, le CSE n'est pas initialisé et ne sera pas appelé pour l'application des paramètres. Pour rappel, le CSE Registre est toujours exécuté avant tous les autres CSE (voir section 2.13).

La référence [8] liste les principaux CSE et leur GUID associé.

Pour qu'un CSE soit chargé et activé par le moteur `GPSvc` lors de l'application d'une GPO, il faut que son GUID soit référencé, au niveau de l'objet GPO dans l'annuaire, dans l'attribut `gPCMachineExtensionNames` pour la partie ordinateur ou l'attribut `gPCUserExtensionNames` pour la partie utilisateur.

Ainsi, lorsqu'une GPO est créée, celle-ci est vide : la version est initialisée à `0` dans l'attribut `versionNumber` et dans le fichier `gpt.ini` et les attributs `gPCMachineExtensionNames` et `gPCUserExtensionNames` sont absents de la définition de la GPO. Il est ensuite de la responsabilité de l'éditeur de GPO de créer ces attributs ou de les mettre à jour en fonction des paramètres définis par la GPO.

Note : il n'est pas rare de voir dans ces attributs des références à des CSE qui sont ne sont plus nécessaires. En effet, les éditeurs de GPO ont tendance à ne pas supprimer la référence au CSE lorsqu'un paramètre est supprimé de la GPO.

Les CSE étant chargés par le moteur `GPsvc` et étant exécutés dans le contexte `LocalSystem`, il est nécessaire d'inventorier les CSE :
— enregistrés dans un système sous la clé `GPExtension` ;
— activés dans les GPO à travers les attributs `gPCMachineExtensionNames` ou `gPCUserExtensionNames`.

2.6 Modes d'application des GPO

Il existe trois modes d'application des GPO : *Synchronous Foreground*, *Asynchronous Foreground* et *Background*.

Les deux modes *Foreground* sont activés uniquement au démarrage/arrêt du système ou à l'ouverture/fermeture d'une session d'un utilisateur. Le mode *Background* est, quant à lui, activé lors des opérations périodiques d'application des GPO ordinateur ou utilisateur. Les CSE ayant la

propriété `NoBackgroundPolicy`[6] ne sont pas appelés dans le mode *Background*. Leurs paramètres ne sont donc appliqués qu'au démarrage/arrêt du système ou à l'ouverture/fermeture des sessions utilisateur.

Dans le mode *Synchronous Foreground*, l'application des GPO doit être terminée pour procéder aux opérations d'authentification (affichage de la mire authentification ou ouverture de la session utilisateur après son authentification). Ce mode permet de s'assurer que l'application des GPO est terminée avant qu'un utilisateur puisse utiliser la machine ou sa session. Inversement, dans le mode *Asynchronous Foreground*, un utilisateur peut s'authentifier ou commencer à travailler alors que les GPO machine ou utilisateur sont toujours en cours d'application.

Depuis Windows XP, à des fins « d'expérience utilisateur[7] », le mode *Asynchronous Foreground* est activé par défaut. Cependant, dans ce mode, seuls les CSE ayant la propriété `EnableAsynchronousProcessing` (voir section 2.5) positionnée à `1` peuvent être appelés. Si le moteur GPO détecte une modification dans une GPO mettant en œuvre un CSE n'ayant pas cette propriété, il doit réactiver temporairement le mode *Synchronous Foreground* et deux démarrages sont nécessaires :

— au premier démarrage, en mode *Asynchronous Foreground*, si le moteur GPO détecte une modification (voir section 2.9) dans une GPO où un CSE n'a pas la propriété `EnableAsynchronousProcessing`, le CSE est désactivé et ses paramètres ne sont pas appliqués. Le moteur GPO modifie sa configuration pour que le mode *Synchronous Foreground* soit activé au prochain démarrage ;

— au second démarrage, le mode *Asynchronous Foreground* est activé et le CSE peut appliquer ses paramètres. Ce comportement explique pourquoi il est recommandé que les CSE `EnableAsynchronousProcessing` soient regroupés et mis en œuvre dans des GPO où les modifications sont rares.

2.7 Filtrage WMI

Pour permettre d'affiner l'application des GPO, il est possible de définir des « filtres WMI » afin de conditionner l'application d'une GPO en fonction de requêtes WQL[8]. Ces filtres prennent la forme d'objets de classe `msWMI-Som` situés sous le conteneur `CN=SOM,CN=WMIPolicy,CN=System`.

6. Par exemple les CSE *Folder Redirection*, *Disk Quota*, *Remote Desktop USB Redirection*, *Work Folders*, *Deployed Printer Connections*, *Software Installation*, etc.

7. Comprendre ici ne pas « rager » en attendant l'application des GPO.

8. *Windows Management Instrumentation Query Language* : langage de requête similaire au SQL sur des classes WMI.

Note : par défaut, les groupes `Domain Admin` et `Enterprise Admins` ont un contrôle total sur le conteneur et les groupes `Administrators` et `Group Policy Creator Owner` peuvent créer des objets sous ce conteneur.

La classe `msWMI-Som` autorise plusieurs attributs qui définissent un filtre WMI :

— `msWMI-ID`, un GUID, qui identifie de manière unique le filtre WMI ;
— `msWMI-Name` qui permet de nommer le filtre ;
— `msWMI-ChangeDate` et `msWMI-CreationDate` qui datent les opérations de création et de modification du filtre ;
— `msWMI-Parm1` qui stocke la description du filtre ;
— `msWMI-Parm2` qui contient la ou les requêtes WQL.

L'attribut `msWMI-Parm2` est donc particulièrement important, car c'est celui qui contient les requêtes WQL. Sa syntaxe est :

```
<Queries count <@\textcolor{red}{n}@>>;
<@\textcolor{red}{n}@> x [<Language size>;<Namespace size>;
<Query size>;<Language>;
<WMI Namespace>;<Query>]
```

Ce qui donne, par exemple, les requêtes suivantes pour filtrer sur les systèmes Windows 10 dont le nom commence par la lettre 'A' :

```
2;3;10;63;WQL;root\CIMv2;select * from Win32_OperatingSystem where
    Version like "10.0.%";3;10;55;WQL;root\CIMv2;select * from
    Win32_ComputerSystem where Name like 'A%';
```

Pour que le filtre soit validé, il faut que chaque rêquete WQL définie dans le filtre retourne au moins un résultat.

Enfin, un filtre WMI est appliqué à une GPO via le champ `gPCWQLFilter`, dont la syntaxe est :

```
[domain;{<Identifiant GUID filtre>};flags]
```

`flags` n'est pas utilisé et vaut systématiquement `0` ; Le champ `gPCWQLFilter` n'étant pas multivalué, il ne peut donc y avoir qu'un seul filtre WMI associé à une GPO.

2.8 Base d'état des GPO dans le Registre

Afin de garder un état de l'application des GPO, le moteur GPO maintient, dans le Registre, plusieurs clés sous `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy`. L'analyse de ces différentes clés permet de connaître l'état d'application :

liste des GPO appliquées, états des CSE, derniers codes d'erreur, scripts exécutés, etc.

Il faut noter que les bases sont toujours de deux types : une partie machine et une ou plusieurs parties pour les utilisateurs, ceux-ci étant identifiés par leur SID.

Partie Ordinateur	Partie Utilisateur
GroupMembership	<SID>\GroupMembership
History	<SID>\History
Shadow	<SID>\Shadow
Status\GPExtensions	Status\<SID>\GPExtensions
State\Machine	State\<SID>

Tableau 2. Clés de Registre des paramètres d'état.

Note : pour rappel, si le CSE, dans sa configuration (voir section 2.5), active l'option `PerUserLocalSettings`, les paramètres d'état du CSE ne seront pas conservés au niveau de la machine (`HKLM`) mais au niveau du profil utilisateur (`HKU\<User Sid>`).

2.9 Version des GPO

Par souci d'optimisation, lors de l'application des GPO, certains CSE ne sont pas appelés si aucune GPO n'a été modifiée depuis la précédente application.

Pour détecter un changement de paramètres, un système de révision est intégré aux GPO : pour chaque GPO, il existe un numéro de révision pour les paramètres ordinateur et un numéro de révision pour les paramètres utilisateur. Ces numéros doivent être incrémentés par les outils d'édition des GPO à chaque modification des paramètres définis par la GPO.

Dans l'Active Directory, le numéro de révision est stocké dans l'attribut `versionNumber` dans chaque objet GPO. Cet attribut est un entier de 32 bits, les 16 bits de poids faible représentant la révision machine et les 16 bits de poids fort la révision utilisateur.

Dans le répertoire de stockage de la GPO, un numéro de version est également présent : il est indiqué par le paramètre `Version` dans le fichier `gpt.ini` présent à la racine du répertoire de chaque GPO. Ce numéro de version ne conditionne pas l'exécution des CSE, mais permet de détecter un éventuel problème de synchronisation ou de réplication si les numéros de version AD/SYSVOL ne sont pas identiques.

Une version à `0` signifie que la GPO est vide, ce qui est la valeur par défaut lorsqu'une GPO est créée.

La liste des GPO appliquées et leur version est stockée dans la base d'état des GPO (voir section 2.8) sous la clé `History`. La liste des GPO appliquées est maintenue individuellement pour chaque CSE.

Un CSE ne sera pas appelé si toutes les conditions suivantes sont réunies :

- la valeur `NoGPOListChanges` est à `1` dans sa configuration ;
- aucune GPO appliquée n'a été modifiée depuis la précédente application. Ceci est détecté par une différence entre la version de la GPO dans l'annuaire et la version de la GPO sous la clé `History` ;
- la précédente exécution du CSE s'est correctement terminée et n'a pas demandé à être appliquée à nouveau (code `ERROR_OVERRIDE_NOCHANGES` [9]) ;
- la limite `MaxNoGPOListChangesInterval` n'est pas définie ou n'a pas encore été dépassée depuis la précédente exécution ;
- les groupes de sécurité de l'ordinateur ou de l'utilisateur n'ont pas été modifiés, l'état précédent étant stocké sous la clé `GroupMembership` (il s'agit de tous les SID contenus dans le *Token* de sécurité).

2.10 Cache GPO

Avec Windows 8, un cache des GPO appelé *Group Policy Caching* a été mis en place. Ainsi, les GPO appliquées (pour l'ordinateur ou l'utilisateur) sont périodiquement récupérées et mises en cache. Les informations issues de l'annuaire sont stockées dans le Registre sous la clé `Group Policy` (voir section 2.8) :

- `DataStore\Machine` contient les GPO appliquées à la machine ;
- `DataStore\<SID>` contient les GPO appliquées à l'utilisateur référencé par son SID.

Le contenu du répertoire de la GPO est également mis en cache. Le chemin sur le disque est indiqué par l'attribut `FileSysPath` et correspond à `C:\Windows\System32\GroupPolicy\DataStore` pour les GPO de la machine et `%LOCALAPPDATA%\GroupPolicy\DataStore` pour les GPO de l'utilisateur.

Le cache est uniquement utilisé lorsque le mode d'application est *Synchronous Foreground* (voir section 2.6). Dans ce cas, les GPO appliquées sont celles issues du cache.

9. Le code de retour de l'exécution du CSE est stocké dans la valeur `Status` sous la clé `Status\GPExtensions\<GUID_CSE>`.

L'activation du cache est paramétrée par la valeur `EnableLogonOptimization` sous la clé `Software\Policies\Microsoft\Windows\System`.

2.11 Journalisation

Le moteur GPO reporte différents événements via le fournisseur `Microsoft-Windows-GroupPolicy`. Les événements génériques (1002 à 1503) sont stockés dans le journal `System`. En revanche, tous les événements de diagnostics (4000 à 9001) sont stockés dans le journal applicatif `Microsoft-Windows-GroupPolicy/Operational` qui est activé par défaut. L'analyse de ce journal permet de voir toutes les différentes étapes effectuées lors d'une opération d'application des GPO décrite dans la section 2.13.

Il est également possible de demander au moteur GPO de journaliser sous forme de fichiers textes encore plus d'opérations via la valeur `GPSvcDebugLevel` sous la clé `HKLM\Software\Microsoft\Windows NT\CurrentVersion`. Dans ce cas, tous les événements sont consignés dans le fichier `%SystemRoot%\debug\usermode\gpsvc.log`. Une explication sur l'analyse de ce fichier est donnée en [9].

2.12 Sécurité des échanges

La sécurité des échanges entre le client GPO et le serveur GPO (i.e. un contrôleur de domaine) est particulièrement importante. En effet, toute atteinte en intégrité pourrait permettre une exécution de code côté client. La protection doit porter aussi bien sur le protocole LDAP que SMB.

Concernant LDAP, lorsque le moteur GPO se connecte au serveur, il applique les protections suivantes :

- l'option `LDAP_OPT_SIGN` est toujours activée, ce qui force la signature des échanges LDAP ;
- dans le cas d'une application des paramètres ordinateur, seul le SSP Kerberos est initialisé, imposant l'utilisation de Kerberos pour l'authentification LDAP (ce qui est nécessaire car la session authentification `0x3e7` ne contient généralement pas de secret d'authentification pour le SSP NTLM, ce qui produirait une authentification anonyme).

De plus, lors de l'initialisation du SSP, la bibliothèque LDAP de Windows demande systématiquement l'authentification mutuelle en spécifiant l'option `ISC_REQ_MUTUAL_AUTH`. Pour rappel, cette propriété ne peut être

vérifiée qu'avec Kerberos, NTLM étant incapable d'assurer l'authentification mutuelle.

Ainsi, la sécurité des échanges LDAP est assurée, pour le contexte de la machine, par l'authentification exclusive de Kerberos, par l'authentification mutuelle requise et par la signature des échanges. Pour le contexte d'un utilisateur, il existe un risque qu'un attaquant force l'utilisation de NTLM puis usurpe l'identité d'un contrôleur de domaine. Ce risque peut être circonscrit par la mise en œuvre d'une clé de session (ce qui est le cas grâce à la signature LDAP imposée [10]), par l'obligation, côté client, d'utiliser NTLMv2 et par l'application des tout derniers correctifs de sécurité sur les contrôleurs de domaine [11].

> S'assurer que NTLMv2 est bien forcé côté client (paramètre `LmCompatibilityLevel` ≥ 3) et que tous les contrôleurs de domaine sont systématiquement mis à jour.

Concernant SMB, le moteur GPO repose entièrement sur la configuration du système pour la sécurité des échanges. La configuration par défaut active le support de la signature des échanges SMB, sans toutefois l'imposer [12]. Pour l'authentification, Kerberos ou NTLM peuvent être utilisés.

Cette configuration par défaut présente une vulnérabilité importante. En effet, un attaquant capable de s'imiscer dans un échange SMB peut désactiver la signature et modifier les paramètres distribués sous forme de fichier.

Une première solution pourrait consister à imposer la signature côté client via le paramètre `RequireSecuritySignature` [13]. Cependant, ce paramètre affecte tout le trafic SMB et peut poser des problèmes de compatibilité pour le client avec certains serveurs.

Pour éviter ce type d'attaque, Microsoft a introduit les *Hardened UNC Paths* via le correctif `MS15-011` [10], disponible pour les éditions Windows de Vista/Server 2008 à 8.1/Server 2012 R2 et intégré depuis Windows 10/Server 2016. Il devient alors possible de spécifier les cri-

10. Le calcul de la clé de session, utilisée pour la signature LDAP, nécessite de connaître les secrets d'authentification de l'utilisateur.

11. Ces trois mesures devant être appliquées simultanément.

12. Sauf pour les contrôleurs de domaine où la signature SMB est imposée côté client.

13. Sous la clé `HKLM\System\CurrentControlSet\Services\LanmanWorkStation\Parameters`.

tères de sécurité (`RequireMutualAuthentication`[14], `RequireIntegrity` et `RequirePrivacy`) aux connexions SMB initiées par le client.

> Imposer, pour les partages `\\*\NETLOGON` et `\\*\SYSVOL` l'utilisation de Kerberos pour bénéficier de l'authentification mutuelle (`RequireMutualAuthentication=1`) et la signature des échanges (`RequireIntegrity=1`). Sans cette configuration, le client reste vulnérable à une attaque de type *man-in-the-middle* pour SMB. Il s'agit de la configuration par défaut depuis Windows 10/Server 2016.

2.13 Application des GPO

Cette section décrit le processus d'application des GPO par le moteur `GPsvc`. La description du processus est ici succincte et a pour objectif d'expliquer les principales étapes et le rôle des différents paramètres. Une description complète du processus est disponible [6]. En particulier, les subtilités de détection de vitesse de lien ou l'application de GPO inter-domaines (authentification d'un utilisateur d'un domaine sur un ordinateur d'un autre domaine) ne sont pas abordées.

L'application des GPO commence toujours par un déclencheur qui démarre le processus. Cela peut être :

- le démarrage ou l'arrêt de la machine : application des paramètres machine ;
- l'ouverture ou la fermeture de session : application des paramètres utilisateur ;
- un *timer* : application des paramètres machine ou utilisateur ;
- une demande d'application manuelle[15] : application des paramètres machine ou utilisateur.

Le moteur GPO commence alors son travail. Les premières opérations consistent à :

- localiser un contrôleur de domaine (fonction `DsGetDcName`) ;
- charger les CSE en fonction de la configuration du Registre (voir section 2.5) ;
- récupérer le *Distinguished Name* de la machine ou de l'utilisateur (fonction `DsCrackNames`) ;
- identifier le site de la machine (fonction `DsGetSiteName`).

Lors de l'opération suivante, toutes les GPO applicables issues du domaine sont récupérées via des requêtes LDAP sur :

14. Qui consiste à utiliser exclusivement Kerberos pour l'authentification.
15. Ceci est généralement effectué au moyen de la commande `gpupdate`.

— les unités d'organisation (OU), du plus proche de l'objet au plus loin (au sens parent dans l'annuaire) ;
— le domaine ;
— le site.

Ces requêtes LDAP permettent de lire l'attribut `gPLink` (qui indique la liste des GPO liées au conteneur ainsi que, pour chaque GPO liée, l'option de liaison `GPLinkOptions`, voir 2.3) et l'attribut `gPOptions` qui définit les propriétés du conteneur, en particulier s'il est bloquant. Toutes les GPO ayant un lien désactivé (bit `1` positionné dans `GPLinkOptions`) ne sont pas ajoutées à la liste des GPO applicables.

Le *Loopback processing*[16] modifie également la liste des GPO applicables à un utilisateur. Si ce mode est activé, deux modifications sont opérées. La première modification consiste à ajouter à la liste des GPO applicables les parties utilisateurs des GPO appliquées à l'ordinateur sur lequel l'utilisateur s'authentifie. Ces GPO étant ajoutées à la fin de la liste, elles deviennent ainsi plus prioritaires sur les autres GPO.

La seconde modification concerne les GPO applicables issues des unités d'organisation parentes de l'objet *User* de l'utilisateur dans l'annuaire. Deux modes sont possibles :

— *Merge mode* : ces GPO sont ajoutées à la liste des GPO applicables, comme lors d'un traitement sans le *Loopback processing* ;
— *Replace mode* : ces GPO ne sont pas ajoutées à la liste, ce qui revient à ignorer leurs paramètres.

Le *Loopback processing* est principalement destiné à obtenir un effet « kiosque » pour forcer l'application de paramètres sur des profils utilisateur.

À cette liste en provenance de l'annuaire sont ajoutées les GPO locales si ceci est autorisé par la configuration[17].

La génération de cette liste s'accompagne d'un tri pour ordonner les GPO et d'un premier filtrage destiné à supprimer les GPO « bloquées ». L'ordre des GPO est important car il détermine les paramètres qui sont « gagnants » en cas de conflit entre GPO. Ainsi, dans cette liste ordonnée, la première GPO sera la moins prioritaire et la dernière GPO la plus prioritaire.

16. Le *Loopback processing* est contrôlé par la valeur `UserPolicyMode` qui peut prendre les valeurs `0` (désactivé, valeur par défaut), `1` (*Merge mode*) ou `2` (*Replace mode*).

17. Ce qui est le cas par défaut, mais les GPO locales peuvent être ignorées via le paramètre `DisableLGPOProcessing`.

L'ordre de base est le suivant : la LGPO est toujours en premier (c'est donc toujours la moins prédominante) suivie des MLGPO de groupe, MLGPO utilisateur, GPO de site, GPO de domaine et GPO d'unités d'organisation, de la plus loin à la plus proche de l'objet.

Cependant, cette liste de base et cet ordre peuvent être modifiés par les paramètres suivants :

— si un conteneur est marqué comme bloquant, c'est-à-dire qu'il possède l'attribut `gPOptions` à `1`, les GPO « au-dessus » ne s'appliquent pas, sauf les GPO locales et les GPO marquées comme *Enforced*. Ainsi, une GPO *Enforced* ne peut jamais être bloquée ;

— si une GPO est *Enforced*, elle est mise en fin de liste et aucune GPO ne pourra être mise après elle dans la liste. Cela signifie que les GPO *Enforced* appliquées à un site seront toujours les plus prioritaires suivies des GPO *Enforced* appliquées à un domaine et ainsi de suite. Cela inverse la logique de priorité : la GPO la plus éloignée devient la plus prioritaire. Les GPO *Enforced* permettent d'appliquer des paramètres en étant sûr qu'ils ne sont pas modifiés par une GPO de niveau inférieur. Ceci est destiné en particulier aux paramètres de sécurité.

À cette étape, une première liste des GPO applicables est établie. Pour chaque GPO de la liste, une requête LDAP est effectuée sur le nom de la GPO (`CN=<GUID_GPO>`) dans le conteneur `CN=Policies,CN=System`. Ceci permet de récupérer les principaux attributs des GPO (`gPCMachineExtensionNames`/`gPCUserExtensionNames`, `ntSecurityDescriptor`, `gpcWQLFilter`, `gPCFunctionalityVersion`, `flags` et `versionNumber`, voir section 2.1) afin d'effectuer un deuxième filtrage sur la base de ces attributs :

— les GPO doivent avoir la bonne version fonctionnelle (`gPCFunctionalityVersion` doit valoir au moins 2) ;

— la partie de la GPO appliquée doit être activée (ordinateur ou utilisateur). Ceci est déterminé par l'attribut `flags` (voir section 2.1) ;

— la GPO ne doit pas être vide (l'attribut `versionNumber` doit être présent et différent de 0) ;

— le compte pour lequel la GPO est appliquée (machine ou utilisateur) doit avoir le droit étendu `ApplyGroupPolicy` sur l'objet GPO : un contrôle d'accès est effectué sur le descripteur de sécurité de la GPO (attribut `ntSecurityDescriptor`) ;

— si un filtre WMI est présent (attribut `gpcWQLFilter`), celui-ci est récupéré depuis l'annuaire puis les requêtes WMI spécifiées par l'attribut `msWMI-Parm2` sont exécutées via la fonction

`IWbemServices::ExecQuery`. Chaque requête WQL doit retourner au moins un résultat.

Si au moins un de ces points de contrôle échoue, la GPO est retirée de la liste. Après cette étape, la liste définitive des GPO à appliquer est établie.

Les CSE sont ensuite appelés les uns après les autres en commençant toujours par le CSE Registre. Les autres CSE suivront, appelés dans l'ordre de leur GUID.

Pour chaque CSE, la fonction de rappel définie dans le Registre (voir section 2.5) est appelée avec pour paramètres :

— `dwFlags`, qui permet de préciser le contexte utilisateur ou machine (`GPO_INFO_FLAG_MACHINE`), le mode d'exécution (`GPO_INFO_FLAG_BACKGROUND`, `GPO_INFO_FLAG_ASYNC_FOREGROUND`) et divers autres paramètres (`GPO_INFO_FLAG_SLOWLINK`, `GPO_INFO_FLAG_FORCED_REFRESH`, etc.) ;

— `pChangedGPOList`, qui indique la liste des GPO à appliquer, c'est-à-dire celles qui ont été modifiées (la détection se fait par une modification du numéro de version, voir section 2.9) ;

— `pDeletedGPOList`, qui indique la liste des GPO supprimées pour que, si applicable, le CSE supprime les paramètres précédemment appliqués. Cette liste est établie en prenant la liste des GPO appliquées précédemment pour le CSE et en retirant les GPO devant être appliquées.

Chaque CSE réalise alors séquentiellement son travail d'application ou de suppression des paramètres.

2.14 CSE Registre

Parmi tous les CSE, celui permettant d'écrire des valeurs dans le Registre (appelée simplement « CSE Registre ») occupe une place particulière [18]. En effet, énormément de paramètres définis par les GPO sont des modifications dans le Registre et c'est ce CSE qui a la responsabilité de les positionner.

Le CSE Registre est toujours exécuté en premier et, en cas d'échec lors de son exécution, les CSE ayant la propriété `RequiresSuccessfulRegistry` ne seront pas activés (voir section 2.5).

Les modèles d'administration sont sans doute les éléments les plus utilisés dans les GPO. Ceux-ci sont un ensemble de paramètres défini

18. Le GUID de ce CSE est `{35378eac-683f-11d2-a89a-00c04fbbcfa2}`.

par des fichiers `.admx` qui font la correspondance entre les composants de l'éditeur graphique et les valeurs à positionner dans le Registre. Les chaînes de caractères de l'éditeur graphique sont, quant à elles, indiquées dans des fichiers `.adml`.

La syntaxe des fichiers ADMX/ADML est décrite dans [2, 3]. De nombreux éditeurs de logiciels fournissent des modèles d'administration pour leur produit (par exemple pour Chromiun, Adobe Reader, etc.).

Cependant, certains paramètres des modèles d'administration ne peuvent pas être appliqués uniquement via une modification dans le Registre. Le modèle d'administration indique alors, via l'attribut `clientExtension` dans le fichier `.admx`, un CSE qui sera chargé de réaliser des opérations complémentaires. Le GUID de ce CSE est alors ajouté, en plus de celui du CSE Registre, à l'attribut `gPCMachineExtensionNames` ou `gPCUserExtensionNames`.

C'est par exemple le cas pour l'activation des protections basées sur la virtualisation. Le modèle d'administration `DeviceGuard.admx` positionne diverses valeurs sous la clé `HKLM\SOFTWARE\Policies\Microsoft\Windows\DeviceGuard`. Mais, en complément, le CSE « *Device Guard Group Policy CSE*[19] » est activé (appel de la fonction `ProcessVirtualizationBasedSecurityGroupPolicy()` de la bibliothèque `dggpext.dll`). Cette fonction a notamment pour rôle de vérifier la compatibilité matérielle avant d'activer la fonctionnalité.

D'autres éléments d'une GPO utilisent également le CSE Registre pour appliquer leurs paramètres. C'est le cas pour AppLocker (`HKLM\Software\Policies\Microsoft\Windows\SrpV2`) ou le pare-feu Windows (`HKLM\Software\Policies\Microsoft\WindowsFirewall`).

Les valeurs à modifier dans le Registre sont stockées dans un fichier dénommé `Registry.pol` dont le format est donné en [7].

3 Audit de la sécurité des GPO

3.1 Droits sur les GPO

Les droits d'accès à une GPO sont donnés par le descripteur de sécurité de son objet dans l'annuaire. Comme pour tous les objets de l'annuaire, le descripteur de sécurité est contenu dans l'attribut `nTSecurityDescriptor`.

Deux types de droits peuvent être spécifiés pour un objet GPO :

19. Dont le GUID est `{f312195e-3d9d-447a-a3f5-08dffa24735e}`.

— ceux standards du modèle de sécurité de Windows (`WRITE_DAC`, `WRITE_OWNER`, `DELETE`, `READ_CONTROL` et `SYNCHRONIZE`) ;
— ceux spécifiques à un objet Active Directory (en particulier `DS_WRITE_PROP` et `DS_CONTROL_ACCESS`).

Pour éviter une gestion complexe des droits des GPO, les outils d'édition de GPO définissent des « modes d'édition » qui sont ensuite traduits en droits sur l'objet GPO :

— *Read* : ce mode applique les droits `READ_CONTROL` (lire les droits d'accès), `DS_LIST` (énumérer les objets fils) et `DS_READ_PROP` (lire toutes les propriétés). Il permet de lire les paramètres de la GPO, mais pas de les modifier ;
— *Edit settings* : ce mode applique les droits `DS_CREATE_CHILD` (créer un objet fils), `DS_DELETE_CHILD` (supprimer un objet fils) et `DS_WRITE_PROP` (écrire toutes les propriétés). Ce mode permet l'édition des paramètres de la GPO ;
— *Apply GPO* : ce mode autorise le droit `DS_CONTROL_ACCESS` qui positionne le droit étendu (*Extended Right*) `ApplyGroupPolicy`[20]. Ce droit est vérifié par le moteur côté client pour autoriser l'application de la GPO sur l'ordinateur ou l'utilisateur (cf. section 2.13).

Il faut vérifier et valider les droits sur le conteneur `CN=Policies,CN=System` dans l'Active Directory. Par défaut, les droits sur le conteneur `Policies` (extrait partiel ne concernant que les ACE explicites, c'est-à-dire non héritées) sont :
— *Authenticated users* : lecture (`READ_CONTROL`, `DS_LIST`, `DS_READ_PROP`, `DS_LIST_OBJECT`) ;
— *System* : tous les droits (contrôle total) ;
— *Domain Admins* : tous les droits sauf `DELETE`, qui permettrait de supprimer l'objet (le conteneur `Policies`) et `DS_DELETE_TREE`, qui permettrait de supprimer l'objet et toutes les GPO ;
— *Group Policy Creator Owners* : `DS_CREATE_CHILD` qui permet de créer des objets fils.

Ces droits par défaut permettent de comprendre le rôle du groupe *Group Policy Creator Owners*. Celui-ci accorde à ses membres les droits nécessaires pour créer des GPO : création d'un objet GPO dans l'Active Directory et création d'un sous-répertoire dans le répertoire `SYSVOL` (voir section 3.3). Lors de la création, le créateur devient propriétaire de l'objet

20. Dont le GUID est `{edacfd8f-ffb3-11d1-b41d-00a0c968f939}`.

dans l'AD et une ACE explicite est positionnée pour lui donner tous les droits, sauf le droit `DS_CONTROL_ACCESS`.

Note : le droit de créer une GPO n'est pas suffisant pour appliquer des paramètres à un ordinateur ou un utilisateur. Pour cela, il est nécessaire de posséder le droit `DS_WRITE_PROP` de attribut `gPLink`[21] sur un conteneur (domaine, unité d'organisation ou site) parent de l'objet cible.

3.2 Points de contrôle

Au niveau des propriétés des GPO (objets de classe `groupPolicyContainer`, section 2.1), on doit s'assurer que :
- le niveau fonctionnel est toujours 2 (attribut `gPCFunctionalityVersion`) ;
- les répertoires des GPO sont bien dans le SYSVOL et de la forme `\\<domain.tld>\SYSOL\Policies\<GUID_GPO>` (attribut `gPCFileSysPath`) ;
- les CSE sont tous identifiés et connus (attributs `gPCMachineExtensionNames` et `gPCUserExtensionNames`).

Au niveau des filtres WMI (objets de classe `msWMI-Som`, voir section 2.7), toutes les requêtes WQL doivent être revues régulièrement (attributs `msWMI-Parm2`).

3.3 Droits NTFS des GPO

Comme vu dans la section 2.2, la majorité des paramètres d'une GPO sont stockés dans le répertoire de stockage de la GPO. Les droits d'accès NTFS sur ce répertoire sont donc tous aussi importants, car ils autorisent ou non l'accès ou la modification des fichiers contenant les principaux paramètres d'une GPO.

21. Dont le GUID est `{f30e3bbe-9ff0-11d1-b603-0000f80367c1}`.

Les répertoires de stockage des GPO étant sous le répertoire `SYSVOL\Policies`, il faut vérifier et valider les droits sur ce répertoire. Par défaut, les droits sont :

- *CREATOR OWNER*, *Administrators* et *System* : tous les droits (contrôle total) ;
- *Authenticated users* et *Servers operators* : lecture ;
- *Group Policy Creator Owners* : modification (qui permet en particulier de créer des sous-répertoires) ;
- *Administrators* : modification, y compris des permissions.

Les droits sur le répertoire `SYSVOL\scripts` [22] doivent également être vérifiés. Ce répertoire contient en particulier les scripts déclarés via l'attribut `scriptPath` d'un profil utilisateur et exécutés à l'ouverture de session. Les droits sont identiques à ceux du répertoire `SYSVOL`, à l'exception de ceux du groupe *Group Policy Creator Owners*, où aucun droit n'est accordé.

Pour éviter des situations inutilement complexes, le moteur d'édition des GPO synchronise automatiquement les droits entre ceux des objets GPO dans l'Active Directory et ceux des répertoires du SYSVOL. Le principe de synchronisation est le suivant :

- les droits de référence d'une GPO sont ceux de son objet dans l'Active Directory ;
- pour chaque répertoire d'une GPO, l'héritage NTFS est supprimé et des droits NTFS explicites sont positionnés ;
- des droits NTFS explicites sont créés depuis ceux de l'objet Active Directory avec le principe de conversion donné par le tableau 3.

Il est important de noter que seules les ACE « simples » sont converties et que les *Object ACEs* sont tout simplement ignorées, comme par exemple celles autorisant le droit étendu `ApplyGroupPolicy`.

Afin d'éviter une éventuelle désynchronisation entre les droits AD et NTFS, l'éditeur des GPO (i.e. `gpmc.msc`) vérifie que ceux-ci soient bien synchronisés. Cette vérification est mise en œuvre par la fonction `IsACLConsistent()` de l'objet COM `GpMgmt.gpm` [23]. En cas de désynchro-

22. Pour rappel, ce répertoire est partagé sous le nom `NETLOGON`.

23. Il s'agit ici de la référence `AppId` de l'objet COM. Ce composant COM est installé avec les outils d'administration des GPO. Ces outils peuvent être installé avec la commande `Add-WindowsCapability -Online -Name Rsat.GroupPolicy.Management.Tools~~~~0.0.1.0` pour les éditions bureaux de Windows ou `Add-WindowsFeature RSAT-ADDS` pour les éditions serveurs.

Droits AD	Droits NTFS
DS_READ_PROP & DS_LIST	FILE_READ_DATA FILE_LIST_DIRECTORY FILE_READ_ATTRIBUTES FILE_READ_EA FILE_EXECUTE
DS_WRITE_PROP	FILE_WRITE_DATA FILE_APPEND_DATA FILE_WRITE_EA FILE_WRITE_ATTRIBUTES FILE_ADD_FILE FILE_ADD_SUBDIRECTORY
DS_CREATE_CHILD	ADD_SUBDIRECTORY FILE_ADD_FILE
DS_DELETE_CHILD	FILE_DELETE_CHILD

Tableau 3. Table de conversion droits AD vers NTFS.

nisation, l'éditeur GPO propose de resynchroniser les droits : ceux à la racine du répertoire de la GPO sont effacés et recréés depuis les droits de l'objet GPO dans l'Active Directory en utilisant le même principe de conversion vu ci-dessus.

Cependant, la vérification effectuée par l'édition ne porte que sur les droits de la racine du répertoire de la GPO et il peut subsister des problèmes de droit sur les sous-répertoires. Si un fichier ou un sous-répertoire possède des droits différents, ceux-ci ne sont pas détectés.

Le script suivant permet d'appeler manuellement une vérification sur toutes les GPO et de resynchroniser les droits le cas échéant :

```
$domain = Get-ADDomain
$gpm = New-object -ComObject "GpMgmt.gpm"
$const = $gpm.GetConstants()
$dom = $gpm.GetDomain($domain.DNSRoot, "", $const.UseAnyDC)

$crit = $gpm.CreateSearchCriteria()
$gpo = $dom.SearchGPOs($crit)
$som = $dom.GetSOM("")

$gpo | ForEach-Object {
   $name = $_.DisplayName
   $c = $_.IsACLConsistent()
   Write-Host -NoNewline $name" - "
   if ($c -eq $FALSE) {
      Write-Host -ForegroundColor Red "Error"
      $_.MakeACLConsistent()
      Write-Host -ForegroundColor Yellow " $name was corrected"
   } else {
      Write-Host -ForegroundColor Green "Ok"
   }
}
```

Vérifier régulièrement la synchronisation des droits des GPO entre l'Active Directory et le SYSVOL.

Pour les GPO locales, les droits sur les répertoires `%SystemRoot%\System32\GroupPolicy` et `%SystemRoot%\System32\GroupPolicyUsers` doivent être ceux par défaut, soit, en syntaxe SDDL[24] : `D:PAI(A;;0x1200a9;;;AU)(A;OICIIO;GXGR;;;AU)(A;;FA;;;BA)(A;OICIIO;GA;;;BA)(A;;FA;;;SY)(A;OICIIO;GA;;;SY)`.
De plus, aucune ACE explicite ne doit être définie sur les sous-répertoires ou les fichiers.

Ces droits accordent la lecture et l'exécution aux utilisateurs authentifiés (pour appliquer les GPO locales) et le contrôle total au groupe *Administrators* (pour pouvoir modifier les GPO locales) et à *LocalSystem*.

3.4 Outil `GPOCheck`

L'outil `GPOCheck` permet de vérifier tous les droits d'un répertoire `SYSVOL` en vérifiant les points suivants :

- que les droits du répertoire racine contenant les GPO dans le `SYSVOL` (`\Policies`) soient ceux par défaut (voir section 3.1) ;
- pour chaque sous-répertoire de GPO :
  - que l'héritage NTFS soit désactivé,
  - que les droits soient ceux attendus de la conversion de l'objet GPO associé dans l'Active Directory ;
- pour chaque fichier ou sous-répertoire de SYSVOL :
  - que l'héritage NTFS soit bien activé,
  - que les ACE soient bien issues de celles du répertoire racine de la GPO,
  - que les ACE héritées soient bien identiques à celles du répertoire racine de la GPO.

Ainsi, `GPOCheck` permet de détecter tous droits NTFS incohérents non détectés par l'éditeur GPO et qui permettrait de modifier un paramètre d'une GPO directement via l'édition d'un fichier.

Attention, ne sont vérifiés ici que les problèmes de désynchronisation de droits entre l'Active Directory et le répertoire `SYSVOL`. En particulier, les droits dangereux positionnés explicitement dans l'Active Directory

24. *Security Descriptor Definition Language*.

devront être audités séparément, par exemple au moyen d'outils de type chemins de contrôle.

4 Audit du contenu des GPO

4.1 Paramètres des GPO

L'analyse des paramètres définis par une GPO se révèle vite compliquée car chaque CSE définit son propre format de stockage des paramètres ainsi que l'emplacement de stockage associé. Certains paramètres sont stockés sous forme d'objet dans l'Active Directory (GPC), d'autres sous forme de fichiers dans le répertoire de stockage de la GPO (GPT) (voir section 2.2).

Voici quelques exemples pour illustrer cette complexité :

— les paramètres de type *Security Settings* (*User Rights Assignments*, *Security Options*, etc.) sont stockés sous forme de fichier texte (`\Machine\Microsoft\Windows NT\SecEdit\GptTmpl.inf`) ;

— les paramètres liés à la politique de la journalisation avancée (*Advanced Audit Policy Configuration*) sont stockés sous forme de fichier CSV (`\Machine\Microsoft\Windows NT\Audit\audit.csv`) ;

— les paramètres de restrictions logicielles (SRP, AppLocker) ou du pare-feu intégré de Windows sont stockés sous forme de fichier binaire (`\Machine\Registry.pol`) (voir section 2.14) ;

— les définitions des scripts à exécuter (Startup, Shutdown, Logon, Logoff) sont stockées sous forme de fichier texte à section (`\Machine\Scripts\scripts.ini`) ;

— les paramètres de « Préférences » sont stockés sous forme de fichier XML (`\Machine\Preferences`) avec un sous-répertoire et fichier différent par type de paramètres ;

— les *Wireless Network Policies* sont stockées sous forme d'objets dans l'annuaire (`CN=<Policy name>, CN=IEEE80211,CN=Windows,CN=Microsoft,CN=Machine, CN=<GUID_GPO>,CN=Policies,CN=System`)

L'analyse des paramètres devrait donc nécessiter de développer un *parseur* pour chaque type d'objet AD ou de fichier (binaire, texte, CSV, XML, etc.).

Heureusement, le moteur GPO permet d'exporter, et ce de manière normalisée, les paramètres d'une GPO. Le premier type d'export est au format HTML et est utilisé en particulier pour l'affichage des paramètres

d'une GPO dans l'éditeur graphique (`gpmc.msc`). Le deuxième type d'export est au format XML, format qui présente l'avantage de pouvoir se traiter et s'analyser facilement.

Pour information, le jeu de stratégies résultant (RSoP) peut aussi être exporté au format HTML (`gpresult /H`) ou XML (`gpresult /X`).

4.2 Outil `gpo2sql`

`gpo2sql` est un outil permettant de convertir les fichiers XML issus d'un export de GPO en tables SQL. L'objectif est de simplifier et d'automatiser l'analyse des paramètres de GPO via des requêtes SQL.

Deux types d'information sont à convertir depuis un fichier XML :
- les informations génériques : identifiant, date de modification, descripteur de sécurité, révision, etc. ;
- les données de chaque CSE, c'est-à-dire les paramètres de la GPO.

Les informations génériques sont communes à toutes les GPO et leur conversion ne pose pas de problème particulier : chaque information devant être convertie est toujours située au même `XPATH` du fichier XML.

En revanche, les paramètres d'une GPO sont regroupés par CSE qui les met en œuvre. Dans le fichier XML, cela prend la forme de sections `<ExtensionData>`. La structure XML des paramètres est alors propre à chaque CSE. `gpo2sql` reconnaît les principaux CSE et convertit les données associées dans une ou plusieurs tables SQL suivant la structure des données issues du fichier XML.

Registry Comme vu dans la section section 2.14, le CSE Registre permet de positionner des valeurs dans le Registre. L'export XML d'un paramètre donne le résultat suivant :

```
<ExtensionData>
<Extension xmlns:q2="http://www.microsoft.com/GroupPolicy/Settings/
    Registry" xsi:type="q2:RegistrySettings">
  <q2:Policy>
    <q2:Name>LSA Protection</q2:Name>
    <q2:State>Enabled</q2:State>
    <q2:Explain>Enable LSA protection</q2:Explain>
    <q2:Supported>At least Windows Server 2012 R2</q2:Supported>
    <q2:Category>MS Security Guide</q2:Category>
  </q2:Policy>
</Extension>
<Name>Registry</Name>
</ExtensionData>
```

`gpo2sql` va convertir les paramètres du CSE dans une table `registry_policy`. L'exemple ci-dessus produira ainsi l'entrée suivante :

```
id_gpo: <identifiant de la GPO dans la table gpo>
name: LSA Protection
state: Enabled
explain: Enable LSA protection
supported: At least Windows Server 2012 R2
category: MS Security Guide
```

Il est important de noter que les chemins des clés de Registre et les valeurs ne sont pas présentes dans l'export XML mais convertis en paramètres textuels suivant les modèles d'administration. Cependant, si le modèle d'administration n'est pas présent, les chemins et les valeurs sont indiqués de manière génétique via des balises XML `<KeyPath>` et `<Value>`. L'exemple suivant reprend l'exemple précédent, mais sans la définition du modèle d'administration. Ces paramètres sont alors mis dans une table `registry_setting`.

```
<ExtensionData>
<Extension xmlns:q2="http://www.microsoft.com/GroupPolicy/Settings/
    Registry" xsi:type="q2:RegistrySettings">
  <q2:RegistrySetting>
    <q2:KeyPath>SYSTEM\CurrentControlSet\Control\Lsa</q2:KeyPath>
    <q2:AdmSetting>false</q2:AdmSetting>
    <q2:Value>
      <q2:Name>RunAsPPL</q2:Name>
      <q2:Number>1</q2:Number>
    </q2:Value>
  </q2:RegistrySetting>
</Extension>
</ExtensionData>
```

Security Le CSE *Security* permet de gérer de nombreux paramètres liés à la sécurité. Ceux-ci sont stockés dans le fichier `GptTmpl.inf` et regroupés par sections : `[Kerberos Policy]`, `[Privilege Rights]`, `[Registry Value]`, `[Service General Settings]`, etc. Notons que les options de l'éditeur de sécurité sont issues de la base de paramètres stockée dans le registre sous la clé `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SecEdit\Reg Values`.

Si la GPO est exportée au format XML, la partie correspondant à ce CSE est donnée dans le listing suivant :

```
<ExtensionData>
  <Extension xmlns:q1="http://www.microsoft.com/GroupPolicy/Settings
      /Security" xsi:type="q1:SecuritySettings">
    <q1:SecurityOptions>
```

```
      <q1:KeyName>MACHINE\System\CurrentControlSet\Control\Lsa\
          LmCompatibilityLevel</q1:KeyName>
      <q1:SettingNumber>1</q1:SettingNumber>
      <q1:Display>
        <q1:Name>Network security: LAN Manager authentication level<
            /q1:Name>
        <q1:DisplayString>Send LM &amp;amp; NTLM - use NTLMv2 session
            security if negotiated</q1:DisplayString>
      </q1:Display>
    </q1:SecurityOptions>
  </Extension>
  <Name>Security</Name>
</ExtensionData>
```

`gpo2sql` va mettre tous les paramètres du CSE *Security* dans diverses tables `security_xxx` où `xxx` représente le type de paramètre de sécurité (*account*, *security options*, *eventlog*, *system services*, etc.). Ainsi, l'exemple ci-dessus sera converti par l'entrée suivante dans la table `security_options` :

```
id_gpo: <identifiant de la GPO dans la table gpo>
keyname: MACHINE\System\CurrentControlSet\Control\Lsa\
    LmCompatibilityLevel
setting: 1
type: Number
display_name: Network security: LAN Manager authentication level
display_string: Send LM &amp;amp; NTLM - use NTLMv2 session security if
    negotiated
```

Enfin, pour identifier toutes les GPO qui définissent le paramètre `LmCompatibilityLevel` ainsi que sa valeur, il suffit d'effectuer la requête :

```
SELECT id_gpo, setting
   FROM security
   WHERE keyname LIKE '%LmCompatibilityLevel'
```

Le ménage des GPO peut ensuite commencer...

5 Conclusion

Très utilisées, les GPO sont un outil important des infrastructures reposant sur l'Active Directory. Elles participent grandement à la sécurité des systèmes Windows. Elles doivent cependant faire l'objet d'une attention particulière et les différents points d'audit présentés dans cet article doivent être contrôlés périodiquement.

A Annexe : documentation Microsoft

À travers son initiative *Open Specifications* [1], Microsoft fournit une documentation conséquente sur le fonctionnement des GPO (des paramètres jusqu'à leur application). Parmi tous les documents, on peut citer

- MS-GPOD : Group Policy Protocols Overview
- MS-GPOL : Group Policy : Core Protocol
- Et toute la documentation des différentes extensions :
  - MS-GPAC : Group Policy : Audit Configuration Extension
  - MS-GPDPC : Group Policy : Deployed Printer Connections Extension
  - MS-GPEF : Group Policy : Encrypting File System Extension
  - MS-GPFAS : Group Policy : Firewall and Advanced Security Data Structure
  - MS-GPFR : Group Policy : Folder Redirection Protocol Extension
  - MS-GPIPSEC : Group Policy : IP Security (IPsec) Protocol Extension
  - MS-GPNRPT : Group Policy : Name Resolution Policy Table (NRPT) Data Extension
  - MS-GPPREF : Group Policy : Preferences Extension Data Structure
  - MS-GPREG : Group Policy : Registry Extension Encoding
  - MS-GPSB : Group Policy : Security Protocol Extension
  - MS-GPSCR : Group Policy : Scripts Extension Encoding
  - MS-GPSI : Group Policy : Software Installation Protocol Extension
  - MS-GPWL : Group Policy : Wireless/Wired Protocol Extension

Références

1. Microsoft. Open Specifications. `https://msdn.microsoft.com/en-us/library/dd208104.aspx/`.
2. Microsoft. Group Policy ADMX Syntax Reference Guide. `https://www.microsoft.com/en-us/download/details.aspx?id=7101`.
3. Microsoft. [MS-GPREG] : Group Policy : Registry Extension Encoding. `https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gpreg/`.
4. Microsoft. Creating a Policy Callback Function `https://docs.microsoft.com/fr-fr/previous-versions/windows/desktop/Policy/creating-a-policy-callback-function`.
5. Microsoft. IPsec Policy Creation/Modification `https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gpipsec/fce021aa-4217-40f3-a7bb-c2f2219eeada`.
6. Microsoft. Policy Application `https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gpol/595ec4ac-95eb-4d56-bec6-aed0e47fb202`.
7. Microsoft. Registry Policy Message Syntax `https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gpreg/5c092c22-bf6b-4e7f-b180-b20743d368f5`.
8. Aurélien Bordes. Group Policy Client Side Extension List `https://github.com/aurel26/gpolist`.
9. Ask the Directory Services Team. A Treatise on Group Policy Troubleshooting–now with GPSVC Log Analysis ! `https://blogs.technet.microsoft.com/askds/2015/04/17/a-treatise-on-group-policy-troubleshootingnow-with-gpsvc-log-analysis/`.

10. Microsoft. MS15-011 : Vulnerability in Group Policy could allow remote code execution : February 10, 2015 https://support.microsoft.com/en-us/help/3000483/ms15-011-vulnerability-in-group-policy-could-allow-remote-code-executi.
11. Microsoft. Step-by-Step Guide to Managing Multiple Local Group Policy Objects https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc766291(v=ws.10).