Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

Secrétariat général
de la défense
et de la sécurité nationale

*Agence nationale de la sécurité
des systèmes d'information*

Paris, le 6 décembre 2013

N° DAT-NT-13/ANSSI/SDE/NP

Nombre de pages du document
(y compris cette page) : 14

Note technique

Recommandations pour la mise en œuvre d'une politique de restrictions logicielles sous Windows

Public visé:

Développeur	
Administrateur	✓
RSSI	✓
DSI	✓
Utilisateur	

Informations

Avertissement

Ce document rédigé par l'ANSSI présente les « **Recommandations pour la mise en œuvre d'une politique de restrictions logicielles sous Windows** ». Il est téléchargeable sur le site www.ssi.gouv.fr. Il constitue une production originale de l'ANSSI. Il est à ce titre placé sous le régime de la « Licence ouverte » publiée par la mission Etalab (www.etalab.gouv.fr). Il est par conséquent diffusable sans restriction.

Ces recommandations sont livrées en l'état et adaptées aux menaces au jour de leur publication. Au regard de la diversité des systèmes d'information, l'ANSSI ne peut garantir que ces informations puissent être reprises sans adaptation sur les systèmes d'information cibles. Dans tous les cas, la pertinence de l'implémentation des éléments proposés par l'ANSSI doit être soumise, au préalable, à la validation de l'administrateur du système et/ou des personnes en charge de la sécurité des systèmes d'information.

Personnes ayant contribué à la rédaction de ce document:

Contributeurs	Rédigé par	Approuvé par	Date
BAS, BAI, BSS	BAS	SDE	6 décembre 2013

Évolutions du document :

Version	Date	Nature des modifications
1.0	6 décembre 2013	Version initiale

Pour toute remarque:

Contact	Adresse	@mél	Téléphone
Bureau Communication de l'ANSSI	51 bd de La Tour-Maubourg 75700 Paris Cedex 07 SP	communication@ssi.gouv.fr	01 71 75 84 04

Table des matières

1 Préambule 3

2 Les mécanismes SRP et AppLocker 3

2.1 SRP ou AppLocker ? . . . 3
2.2 Fonctionnement d'AppLocker . . . 4

3 Mise en œuvre d'une stratégie de restrictions logicielles avec AppLocker 5

3.1 Démarche préalable . . . 5
3.1.1 Réaliser un inventaire des applications utilisées . . . 5
3.1.2 Activer et configurer AppLocker sur les configurations . . . 6
3.2 Configuration des règles . . . 7
3.2.1 Créer les règles pour les exécutables . . . 7
3.2.2 Créer les règles pour les scripts et les « installeurs » . . . 9
3.2.3 Créer les règles pour les bibliothèques (optionnel) . . . 10
3.3 Tester les règles mises en place et affiner leur configuration si nécessaire . . . 11
3.4 Maintenir les règles à jour au gré des évolutions des configurations . . . 11

4 Considérations de sécurité fondamentales 12

Annexes 13

A Les variables de chemin d'accès utilisées par AppLocker 13

B Liste non exhaustive des évènements générés par AppLocker 13

1 Préambule

L'intérêt principal des mécanismes de restriction logicielle réside dans la possibilité de restreindre l'exécution des programmes à une liste de programmes dûment autorisés (liste blanche). Le principe d'une liste blanche procure plusieurs avantages :

- une meilleure protection contre les programmes malveillants : en empêchant systématiquement l'exécution des programmes non répertoriés dans la liste, on bloque également ceux susceptibles de contenir un code malveillant, que ce dernier soit connu ou non des bases de signature de l'antivirus ;
- un blocage de l'installation ou de l'utilisation de logiciels indésirables, en particulier ceux qui sont susceptibles de ralentir ou de rendre instables les configurations, et qui dans tous les cas augmentent la surface d'attaque des configurations ;
- un blocage de l'installation ou de l'utilisation de logiciels sans licence.

Sur un système à jour de ses correctifs de sécurité et respectant le principe de séparation des privilèges, l'activation des mécanismes de restriction logicielle augmente sensiblement la maîtrise des configurations.

Enfin, en règle générale, l'activation des mécanismes de restriction logicielle n'engendre pas de ralentissement sur la machine, cette mesure est totalement transparente pour l'utilisateur.

2 Les mécanismes SRP et AppLocker

2.1 SRP ou AppLocker ?

À partir de Windows XP et Windows 2003 server, Microsoft introduit les SRP (*Software Restriction Policies*) permettant la mise en œuvre d'une politique de restrictions logicielles. Grâce aux SRP, il devient possible de restreindre l'exécution de programmes sur la machine de l'utilisateur en définissant une liste blanche à l'aide de règles particulières.

Bien configurées, les SRP offrent une protection efficace. Néanmoins, leur configuration peut être assez lourde à maintenir dans un environnement dynamique, en particulier sur un grand parc de machines. À cet égard, AppLocker qui est une évolution des SRP apporte de nettes améliorations.

Une différence fondamentale entre SRP et AppLocker est le champ d'application des règles. Avec SRP, tous les utilisateurs d'une machine sont impactés indifféremment par les règles. Avec AppLocker, il est possible de cibler un utilisateur précis d'une machine ou d'un domaine.

Pour pouvoir bénéficier d'AppLocker sur les postes utilisateur, il est impératif de disposer de Windows 7 édition Entreprise/Intégrale ou Windows 8 édition Entreprise. Les éditions « professionnelles » de Windows 7 et Windows 8 permettent de configurer des règles AppLocker, mais ces dernières sont inopérantes. Ainsi, AppLocker peut être activé sur les systèmes suivants :

- Windows 7 éditions « Entreprise » et « Intégrale » ;
- Windows 8 édition « Entreprise » ;
- Windows Server 2008 R/2 éditions « Standard », « Entreprise », « Datacenter » et pour les systèmes Itanium ;
- Windows Server 2012 éditions « Standard » et « Datacenter ».

Note : Windows 7 et Windows 8 supportent toujours les SRP, mais si une stratégie configure simultanément des règles SRP et AppLocker, seules les règles AppLocker seront prises en compte.

R1	Pour mettre en œuvre une politique de restrictions logicielles fine, il est préférable d'utiliser Applocker plutôt que SRP .

Note : Dans la suite de ce document, seul l'usage d'AppLocker est abordé.

2.2 Fonctionnement d'AppLocker [1]

Avant de poursuivre, il est nécessaire de comprendre le fonctionnement d'AppLocker, les avantages et inconvénients des différentes règles mais aussi la manière dont elles interagissent entre elles.

AppLocker permet d'« AUTORISER » ou de « REFUSER » à un utilisateur (ou un groupe d'utilisateurs) le lancement de programmes sous différentes formes :

- des exécutables : fichiers .exe et .com ;
- des Windows Installer : fichiers .msi et .mst ;
- des scripts : .ps1 (powershell), .bat, .cmd, .vbs, .js ;
- des bibliothèques : .dll et .ocx.

Pour déterminer si un programme est autorisé à s'exécuter ou non, AppLocker évalue d'abord les règles du type « REFUSER » puis celles du type « AUTORISER ». Il est possible de combiner des règles « AUTORISER » et « REFUSER ». Pour une règle donnée, il peut aussi être défini une ou plusieurs exceptions. Lorsqu'un programme ne fait l'objet d'aucune règle du type « AUTORISER », il est automatiquement bloqué (refus implicite).

Lorsqu'un programme est bloqué par Applocker, l'utilisateur en est informé par un message d'erreur du type : *« Ce programme a été bloqué par une stratégie de groupe, veuillez contacter votre administrateur »*. Ce message peut être personnalisé en modifiant un paramètre de la GPO [2], un lien « Plus d'informations » dont le contenu est paramétrable apparaît. Cette possibilité peut être utilisée par exemple pour rediriger l'utilisateur vers l'ouverture d'un ticket d'incident.

R2	Pour une meilleure lisibilité du comportement d'AppLocker, il est préférable de n'utiliser que des règles du type « AUTORISER » avec si nécessaire des exceptions.

L'autorisation ou le refus d'exécution d'un programme est conditionné à la vérification de règles pour lesquelles trois types différents existent :

- les règles basées sur le chemin d'accès, qui permettent d'autoriser ou de refuser l'exécution de fichiers se trouvant dans le répertoire et les sous-répertoires du chemin. Pour désigner les répertoires classiques du système de fichiers, AppLocker utilise des variables qui sont différentes des variables d'environnement de Windows (voir le tableau de correspondance en annexe A) ;
- les règles basées sur une signature électronique, permettent d'autoriser seulement les fichiers signés par un éditeur donné, et répondant éventuellement à d'autres critères comme le nom du produit, le nom du fichier et sa version ;
- les règles basées sur l'empreinte cryptographique (sha256) d'un fichier, qui n'autorisent que le fichier correspondant à l'empreinte.

1. Ce paragraphe est une synthèse des éléments fournis par Microsoft sur le site technet.microsoft.com que le lecteur est invité à consulter pour tout complément d'informations ou toute précision sur le fonctionnement d'AppLocker.

2. Le paramètre est accessible dans l'arborescence suivante : configuration ordinateur -> stratégies -> modèles d'administration -> composants Windows -> Explorateur Windows-> Définir le lien d'une page web de support.

Les règles basées sur le chemin d'accès offrent une grande souplesse, mais exigent en contre-partie la maîtrise dans le temps du contenu et des autorisations des répertoires associés afin de s'assurer que seuls des programmes légitimes peuvent s'y trouver. En général, la mise à jour d'un logiciel n'oblige pas à modifier les règles existantes.

Les règles basées sur une signature électronique obtenue à l'aide de certificats de confiance offrent quant à elles plus de sécurité et, selon leur configuration, une souplesse à géométrie variable. Les mises à jour des programmes sont en général transparentes.

Les règles basées sur des empreintes offrent le meilleur niveau de sécurité car elles n'autorisent que les fichiers correspondant à l'empreinte cryptographique. Par contre, lors de la mise à jour d'un logiciel, les règles doivent la plupart du temps être modifiées.

De façon analogue aux règles, les exceptions peuvent s'appliquer à un répertoire, à une signature électronique ou à une empreinte de fichier, et ce quel que soit le type de la règle auxquelles elles sont rattachées.

R3	Lorsque cela est possible, il convient d'utiliser des règles basées sur la signature électronique pour autoriser ou refuser l'exécution d'un programme en s'étant assuré au préalable que les certificats et les autorités de certification sont de confiance.

3 Mise en œuvre d'une stratégie de restrictions logicielles avec AppLocker

Les principales étapes pour le déploiement d'une stratégie de restrictions logicielles sont les suivantes :

- réaliser un inventaire des applications utilisées et autorisées sur l'ensemble des machines du domaine Windows ;
- créer les règles pour les applications autorisées à s'exécuter ;
- créer les règles pour les scripts et les installeurs ;
- créer des règles pour les bibliothèques (optionnel) ;
- tester les règles mises en place et affiner leur configuration si nécessaire.

3.1 Démarche préalable

3.1.1 Réaliser un inventaire des applications utilisées

Cette étape consiste à répertorier l'ensemble des applications nécessaires à la réalisation des missions d'une entité. Pour cela, il convient d'établir une liste des logiciels autorisés sur la base de critères bien précis (fonctionnalités, robustesse, mises à jour de sécurité régulières, etc.), qui permettra de définir une (ou plusieurs) configuration(s) de référence. Des logiciels de gestion de parc informatique peuvent être utilisés pour connaître les applications utilisées au sein d'un organisme.

AppLocker possède également une fonctionnalité d'audit qui permet de simuler l'application d'une politique de restriction logicielle sans bloquer l'exécution des programmes. Lorsque cette fonctionnalité est activée, les règles ne sont pas appliquées mais simplement évaluées, et tous les évènements générés sont écrits dans le journal d'AppLocker. Ce dernier peut être consulté dans l'observateur d'évènements de Windows en parcourant l'arborescence de la manière suivante : *Journaux des applications et services -> Microsoft -> Windows -> AppLocker*.

R4	Lorsqu'il n'existe pas d'inventaire exhaustif des applications utilisées dans une organisation, la fonctionnalité d'audit d'AppLocker peut être utilisée pour identifier les applications inconnues.

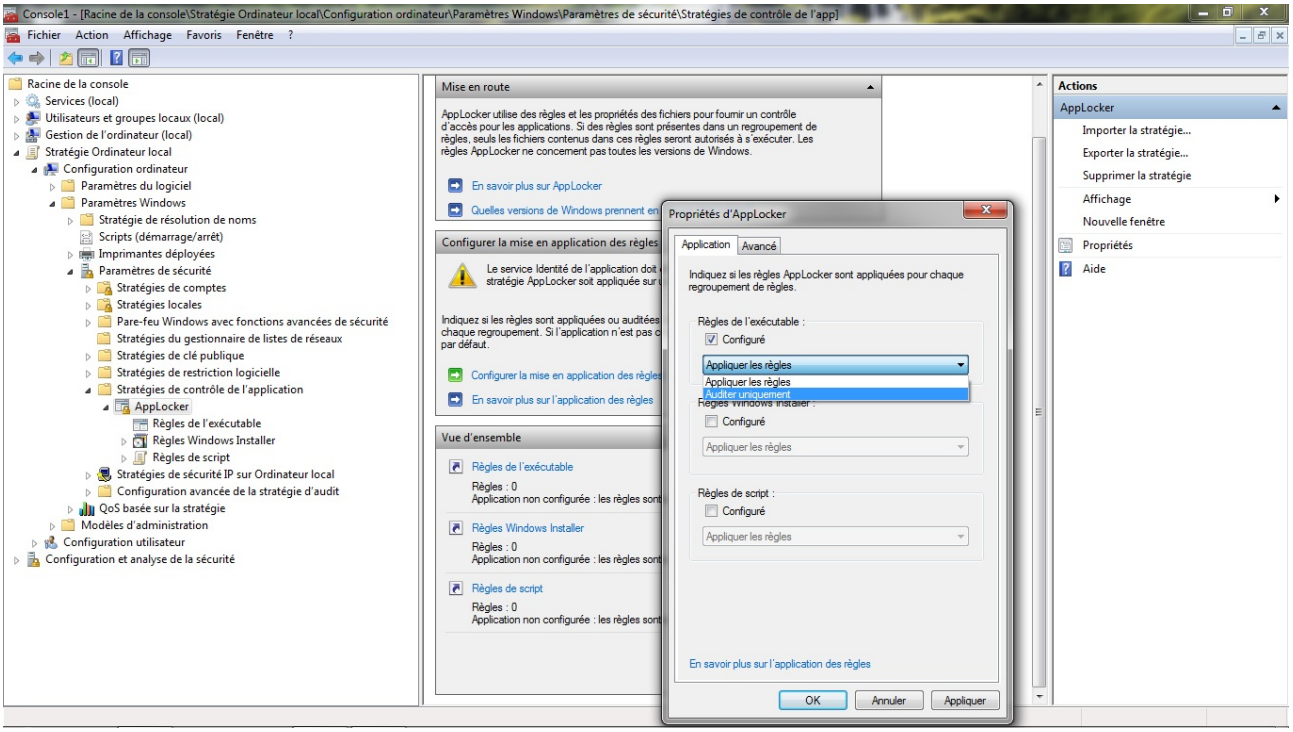

FIGURE 1 – Configuration d'AppLocker en mode audit

3.1.2 Activer et configurer AppLocker sur les configurations

Une fois l'inventaire des applications réalisé, il reste à activer le mécanisme et à créer les règles correspondantes sur la configuration de référence. AppLocker ne fonctionne que si le service Identité de l'application est démarré.

R5	Pour activer AppLocker sur une configuration, le service AppIdSvc - Identité de l'application - doit être configuré pour démarrer automatiquement au lancement du système.

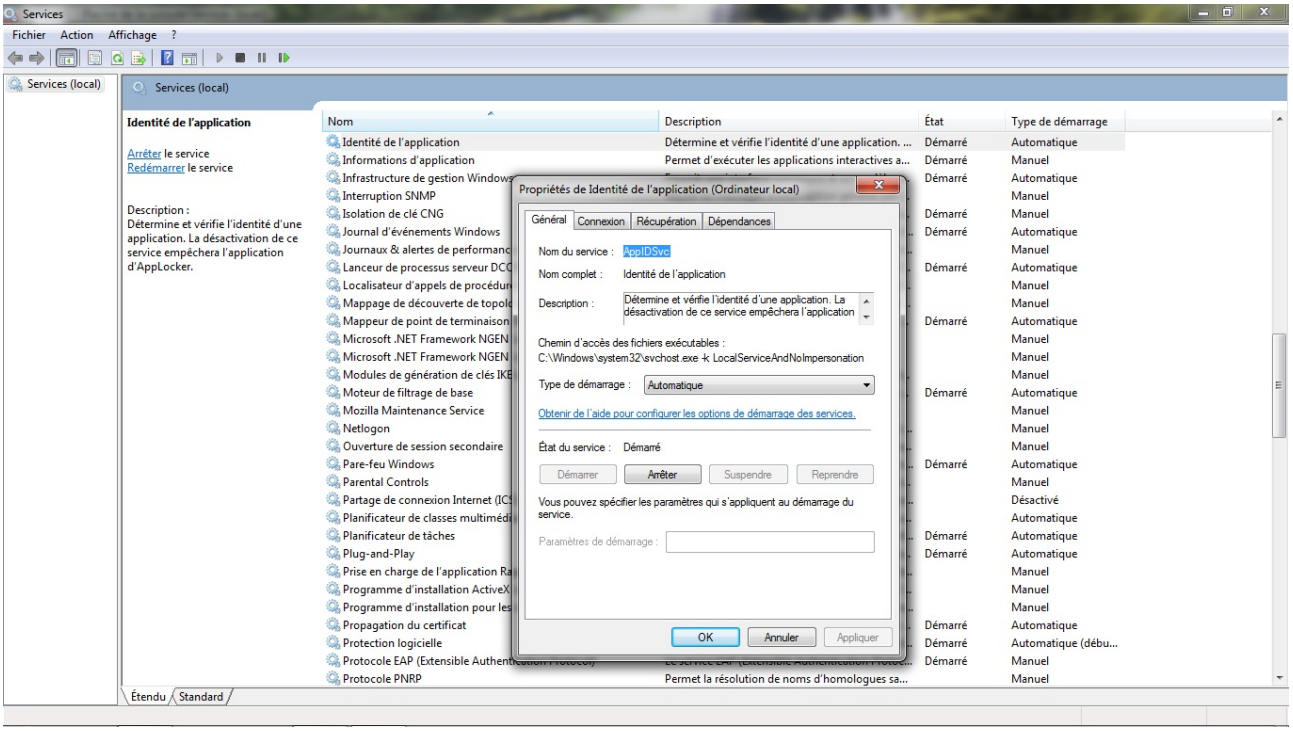

FIGURE 2 – Configuration du service identité de l'application

R6	Les configurations sur lesquelles AppLocker est déployé doivent être utilisées avec un compte utilisateur standard. Les utilisateurs ne doivent en aucun cas disposer de privilèges d'administration locaux.

AppLocker peut être configuré au travers d'une stratégie locale (applicable à une machine donnée) ou bien une stratégie de groupe (Group Policy Object) lorsque les machines sont rattachées à un domaine. Dans ce cas, il est conseillé de tester préalablement la stratégie en mode « audit » sur un échantillon représentatif du parc informatique, puis de l'appliquer progressivement sur l'ensemble du domaine.

3.2 Configuration des règles

3.2.1 Créer les règles pour les exécutables

Cette étape consiste à créer et tester les règles permettant aux utilisateurs d'exécuter les logiciels autorisés dans votre organisation. Si aucune règle n'est définie, l'exécution de tous les programmes sera bloquée, y compris les exécutables système. En pratique cela pourrait se traduire par l'impossibilité d'ouvrir une session sur la machine.

La fonction « créer des règles par défaut » crée automatiquement trois règles de type « chemin d'accès » permettant d'activer AppLocker avec une configuration qui, dans la plupart des cas, sera immédiatement opérationnelle. Il s'agit de :

- une règle autorisant tous les utilisateurs à exécuter les programmes situés dans `c:\Windows` et ses sous-répertoires ;
- une règle autorisant tous les utilisateurs à exécuter les programmes situés dans `c:\Program Files` et ses sous-répertoires ;

– une règle autorisant les administrateurs à exécuter les fichiers depuis tous les emplacements.

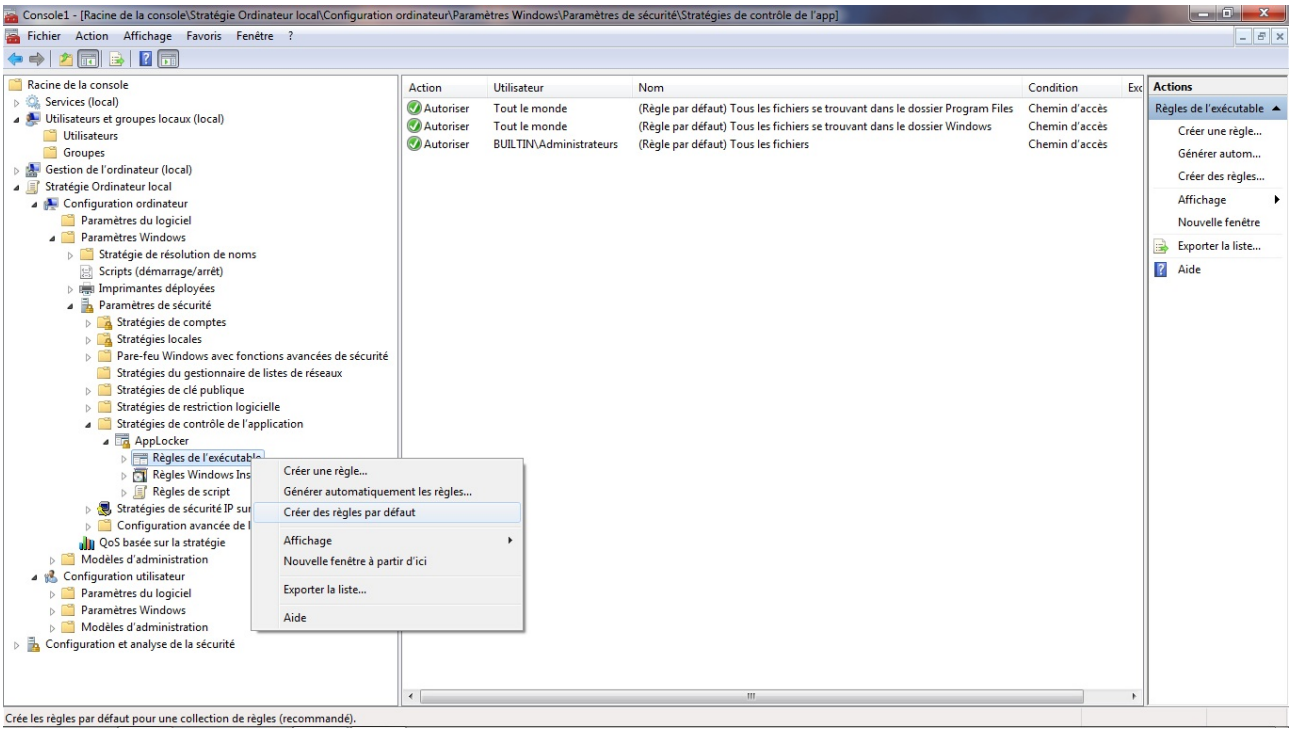

FIGURE 3 – Règles créées par défaut pour les exécutables

Il est à noter que la règle créée par défaut pour `c:\Windows` autorise de fait l'exécution de programmes depuis des emplacements sur lesquels un utilisateur standard dispose des droits d'écriture, par exemple `c:\Windows\temp`, `c:\Windows\tasks` ou encore `c:\Windows\system32\spool\drivers\color`. Ainsi, tout utilisateur standard peut contourner cette règle de façon triviale, en créant ou en copiant un programme dans ces emplacements. Pour remédier à ce problème, il est recommandé de lister l'ensemble des répertoires accessibles en écriture aux utilisateurs standard et d'ajouter des exceptions de type chemin d'accès à la règle par défaut. Un logiciel comme AccessEnum pourra être avantageusement utilisé pour identifier rapidement les répertoires en question.

R7	Lorsqu'une règle de type « chemin d'accès » est utilisée, le dossier spécifié ainsi que ses sous-dossiers ne doivent être accessibles en écriture qu'aux administrateurs et aux entités SYSTEM.

R8	Pour les sous-répertoires de Windows accessibles en écriture par les utilisateurs, la règle créée par défaut pour `c:\Windows` doit être complétée par l'ajout d'exceptions sur ces derniers.

Note : Une méthode alternative est de supprimer la règle par défaut au profit d'une règle de type Editeur n'autorisant que les composants signés « Microsoft Operating System ».

De façon analogue, la règle créée par défaut pour `c:\Program Files` autorise tous les programmes de ce dossier à s'exécuter. Il convient donc de s'assurer que toutes les applications présentes à cet emplacement sont autorisées. Parmi ces applications, il convient aussi de s'assurer qu'aucune n'y a créé

de répertoire accessible aux utilisateurs en écriture.

Néanmoins, sur un parc informatique existant, il est peu probable que toutes les configurations soient homogènes. Aussi, pour une meilleure maîtrise des applications autorisées à s'exécuter et une meilleure lisibilité des règles, il est préférable de supprimer la règle par défaut au profit de règles autorisant explicitement les applications stockées à cet emplacement.

R9	Il est recommandé de supprimer la règle par défaut concernant `c:\Program Files` et de la remplacer par des règles autorisant explicitement chaque application de cet emplacement à s'exécuter.

Note : Une méthode alternative est de supprimer la règle par défaut au profit de règles basées sur les signatures numériques des applications ou, lorsque possible, les empreintes.

On notera la présence d'un assistant permettant de générer automatiquement les règles des exécutables en scannant les dossiers dans lesquels ils se trouvent. L'assistant permet de choisir entre 2 algorithmes :

- création de règles basées sur des signatures numériques le cas échéant, dans le cas contraire l'assistant propose au choix de créer une règle basée sur l'empreinte ou sur le chemin d'accès ;
- création de règles basées exclusivement sur les empreintes des fichiers.

Bien que cet assistant facilite la tâche de création des règles, il convient de s'assurer que les règles ainsi générées ne sont pas trop permissives.

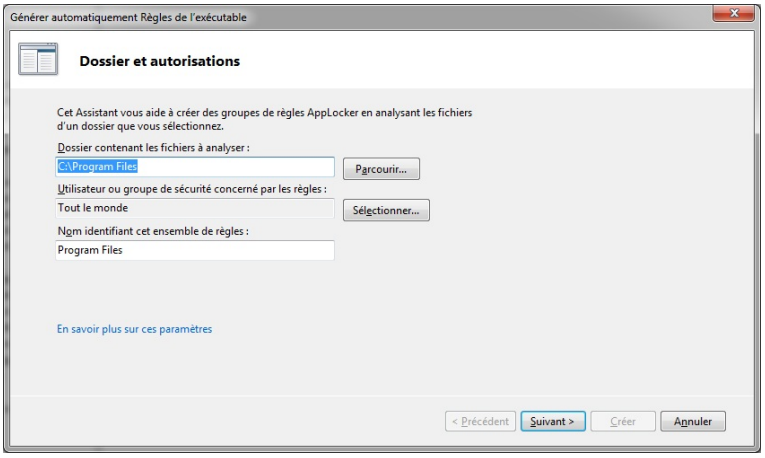

FIGURE 4 – Démarrage de l'assistant de génération automatique de règles

3.2.2 Créer les règles pour les scripts et les « installeurs »

Les règles pour les scripts et les « installeurs » doivent également être configurées. S'agissant des « installeurs », les règles créées par défaut n'autorisent que les fichiers signés ou se trouvant dans `c:\Windows\installer` (non accessible en écriture aux utilisateurs). Sur un réseau bien administré, ces règles ne posent pas de problème particulier.

En ce qui concerne les scripts, les règles par défaut n'autorisent l'exécution de ces derniers que depuis `c:\Windows`, ce qui, comme vu précédemment, peut être contourné.

R10	Supprimez les règles autorisant l'exécution des scripts depuis `c:\Windows`. N'autorisez que des scripts signés numériquement ou désignés explicitement par leur empreinte.

3.2.3 Créer les règles pour les bibliothèques (optionnel)

Par défaut, les règles concernant les bibliothèques ne sont pas appliquées. Un message d'avertissement met en garde l'administrateur concernant un possible impact sur les performances ainsi que des comportements inattendus si les règles ne sont pas correctement définies.

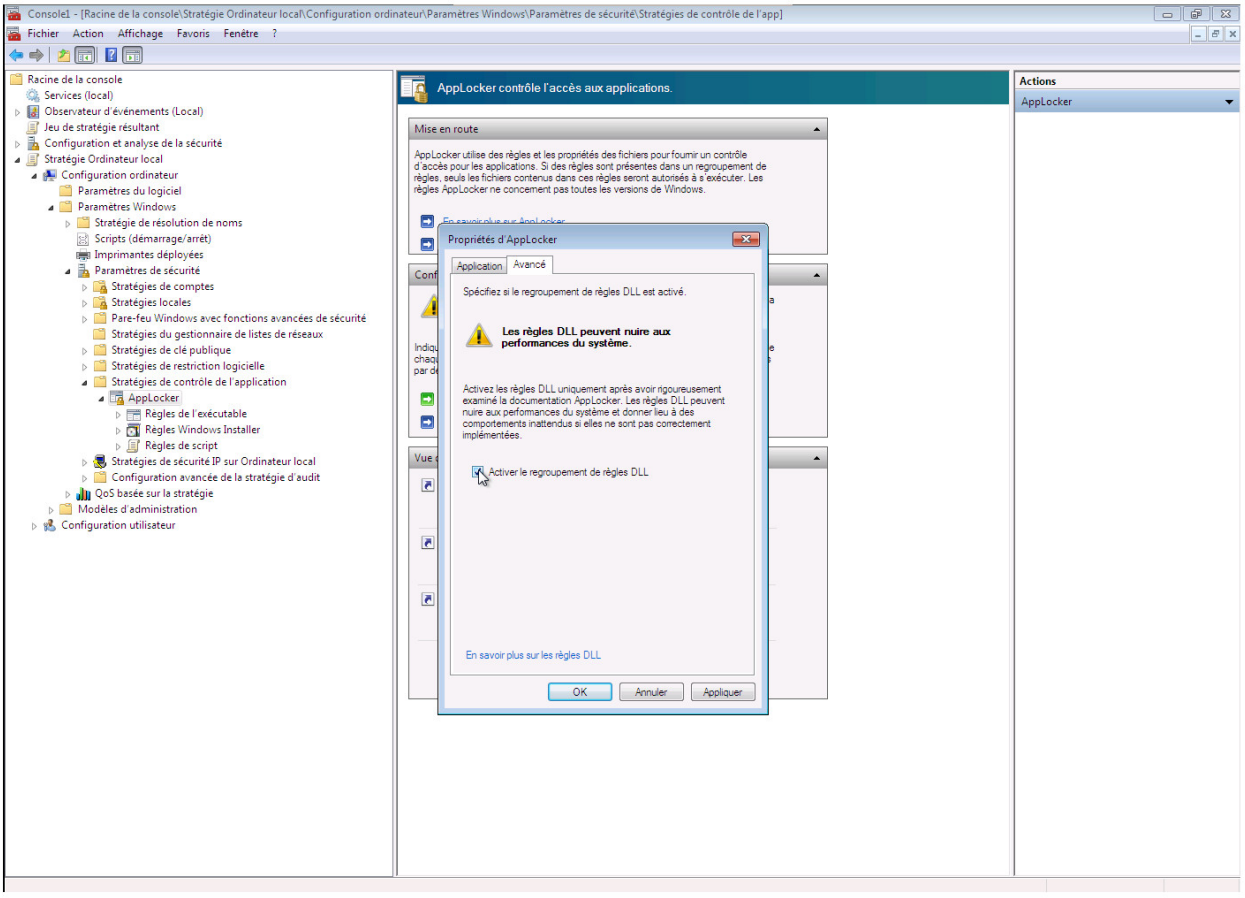

FIGURE 5 – Activation des règles DLL

En fonction des configurations logicielles, l'activation des règles DLL peut effectivement nécessiter l'ajout plus ou moins conséquent de règles spécifiques afin de garantir le bon fonctionnement des applications.

Quoi qu'il en soit, l'activation des règles DLL est toujours à considérer, des vulnérabilités peuvent être exploitées pour charger une bibliothèque contenant un code malveillant.

R11	Dans la mesure du possible, activez les règles concernant les DLL en cochant la case dans l'onglet avancé de la fenêtre de configuration de mise en application des règles.

R12	À l'instar des règles relatives aux exécutables, les règles par défaut doivent être complétées par l'ajout d'exceptions portant sur les sous-répertoires des chemins d'accès qui restent accessibles en écriture aux utilisateurs.

3.3 Tester les règles mises en place et affiner leur configuration si nécessaire

Lorsque les règles ont été créées et testées avec succès sur la configuration de référence, elles doivent être testées en conditions réelles.

Dans un premier temps, afin d'éviter de bloquer les utilisateurs, la configuration en mode audit pourra s'avérer utile. Pour faciliter l'analyse de l'ensemble des évènements générés sur les machines, il est possible de centraliser ces derniers grâce à la fonction de collecte des évènements disponible depuis 2003 R2, Vista SP1 et Windows 2008.

La collecte des évènements peut se faire en créant un abonnement sur la machine prévue à cet effet, ce qui permet en outre de sélectionner seulement les évènements intéressants grâce à un filtre de requête. Une liste non exhaustive des évènements générés par AppLocker est fournie en annexe B.

Il est également possible d'obtenir des statistiques sur les fichiers qui sont (ou seraient) bloqués par une stratégie AppLocker grâce à la cmdlet Powershell *GetAppLockerFileInformation*. Par exemple, la commande suivante permet de connaître le nombre de fois qu'un fichier aurait été bloqué si les règles avaient été appliquées :

```
Get-AppLockerFileInformation -EventLog -Logname c:\...\fichier-log -EventType Audited
-Statistics
```

Dans un deuxième temps, AppLocker pourra être déployé sur une partie du parc informatique, de préférence sur un échantillon représentatif, puis, si les tests sont concluants, étendu à l'ensemble du domaine.

3.4 Maintenir les règles à jour au gré des évolutions des configurations

Il va sans dire que l'installation de nouveaux logiciels, de mises à jour ou correctifs sur le parc informatique doit toujours êtres précédée d'une phase de qualification sur des configurations de référence. En effet ces actions pourraient être à l'origine de blocages ou de comportements anormaux pour les utilisateurs, nécessitant la modification ou l'ajout de règles.

Le choix des règles doit être adapté aux spécificités des logiciels. Il s'agit de trouver le meilleur compromis entre niveau de sécurité et facilité de maintenance. Même lorsque les exécutables sont signés, il n'est pas toujours possible d'écrire des règles adaptatives (il arrive en effet que le numéro de version du logiciel soit inclus dans le champ nom du produit, ce qui empêche d'écrire une règle pérenne).

4 Considérations de sécurité fondamentales

Lors de la configuration d'AppLocker, il faut être concient de certaines limites et possibilités de contournement.

Les composants ActiveX peuvent permettre d'instancier certains programmes dont l'exécution devrait être bloquée par AppLocker (exemple : il est possible de lancer une session RDP en insérant le contrôle Microsoft RDP client dans un document Word).

R13	Pour les programmes dont l'exécution est bloquée par AppLocker, si d'autres composants logiciels permettent de les instancier, ils doivent aussi être bloqués par des règles sur les DLL/OCX associées.

Les règles AppLocker autorisent ou empêchent le lancement des programmes, mais n'exercent aucun contrôle sur le comportement des programmes une fois lancés. En utilisant des fonctions avec des paramètres spéciaux[3], un programme peut lancer des exécutables ou des bibliothèques sans que les règles ne s'appliquent. En particulier, les modules d'extension (greffons, plugins, etc.) de certains logiciels peuvent être exploités par un attaquant pour contourner les restrictions d'Applocker.

R14	Les programmes autorisés à s'exécuter doivent avoir fait l'objet de vérifications pour s'assurer qu'ils ne comportent pas de fonctionnalités susceptibles de permettre un contournement d'Applocker. Les utilisateurs ne doivent pas être autorisés à installer des modules d'extension.

AppLocker ne contrôle pas l'exécution de tous les codes interprétés existants, comme par exemple les scripts Perl ou les macros. Des mesures de sécurité supplémentaires doivent être prises afin de restreindre les possibilités en la matière.

R15	Les logiciels offrant des possibilités de « scripting » doivent être paramétrés de façon à restreindre les possibilités d'exécution des codes interprétés. Dans le cas d'une suite bureautique, outre le paramétrage restreignant l'exécution aux seules macros signées et de confiance, il est recommandé d'empêcher l'utilisation de l'éditeur de macros.

Les applications 16 bits ne sont pas contrôlées par AppLocker car celles-ci sont exécutées dans une machine virtuelle instanciée par le processus NTVDM - NT Virtual Dos Machine - (`c:\Windows\system32\NTVDM.exe`).

R16	Si aucune application 16 bits n'est utilisée dans votre organisation, désactivez la fonction NTVDM avec l'éditeur de stratégie de groupe[4].

3. Lorsque la fonction LoadLibraryEx() est appelée avec le drapeau LOAD_IGNORE_CODE_AUTHZ_LEVEL, les règles AppLocker sont ignorées (voir la page « Security Considerations for AppLocker » sur le site technet.microsoft.com).

4. La fonction peut être désactivée en se rendant dans l'arborescence suivante : Configuration ordinateur -> modeles d'administration -> composants Windows -> compatibilité des applications 16 bits -> Empêcher l'accès aux applications 16 bits.

Annexes

A Les variables de chemin d'accès utilisées par AppLocker

Le tableau ci-dessous indique les variables utilisées par AppLocker pour désigner les principaux chemins d'accès de Windows. La correspondance avec les variables d'environnement couramment utilisées est également indiquée.

Chemin d'accès	Variable d'environnement Windows	Variable AppLocker
Windows	%SystemRoot%	%WINDIR%
System32	%SystemDirectory%	%SYSTEM32%
Lecteur d'installation de Windows	%SystemDrive%	%OSDRIVE%
Répertoire des programmes	%ProgramFiles% et %ProgramFiles(x86)%	%PROGRAMFILES%
Media amovibles	-	%REMOVABLE%
Périphériques de stockage amovibles	-	%HOT%

B Liste non exhaustive des évènements générés par AppLocker

Le tableau ci-dessous recense les identifiants et descriptifs des principaux évènements générés par Applocker, en fonction du type de fichiers exécutables.

Identifiant	Type	Description	Fichiers
8001	Information	La stratégie AppLocker a été correctement appliquée	-
8002	Information	<Fichier> a été autorisé à s'exécuter.	EXE ou DLL.
8003	Avertissement	<Fichier> a été autorisé à s'exécuter mais aurait été bloqué si les règles étaient appliquées.	
8004	Erreur	<Fichier> n'a pas été autorisé à s'exécuter.	
8005	Information	<Fichier> a été autorisé à s'éxécuter.	Script ou Installeur
8006	Avertissement	<Fichier> a été autorisé à s'exécuter mais aurait été bloqué si les règles étaient appliquées.	
8007	Erreur	<Fichier> n'a pas été autorisé à s'exécuter.	