Sécuriser son parc Windows avec le projet modulaire et communautaire SWMB

Olivier De-Marchi, Clément Deiber, Gabriel Pierre André Moreau, David Gras, Philippe Hortolland, Olivier Pavilla, Sébastien Morin

Sécuriser son parc Windows avec le projet modulaire et communautaire SWMB

Olivier De Marchi
Laboratoire LEGI
Domaine Universitaire
1209-1211 rue de la piscine
38 400 Saint-Martin-d'Hères

Clément Deiber
Délégation CNRS Alpes
25 rue des Martyrs
38 000 Grenoble

Gabriel Moreau
Laboratoire LEGI
Domaine Universitaire
1209-1211 rue de la piscine
38 400 Saint-Martin-d'Hères

David Gras
Délégation CNRS Alpes
25 rue des Martyrs
38 000 Grenoble

Philippe Hortolland
Institut de Mécanique et d'Ingénierie
Site ENSAM – Esplanade Arts et Métiers
33 400 Talence

Olivier Pavilla
Centre d'Économie de la Sorbonne
Maison des Sciences Économiques
106-112 boulevard de l'Hôpital
75 013 Paris

Sébastien Morin
Laboratoire DCM
Domaine Universitaire
301 rue de la chimie
38 400 Saint-Martin-d'Hères

Résumé

*Le projet SWMB a été initié en fin d'année 2019. **SWMB** signifie **S**ecure **W**indows **M**ode **B**atch. Il s'agit de simplifier et d'automatiser la sécurisation d'un parc sous Microsoft Windows 10. Dans ce but, le groupe de travail RESINFO a écrit, en PowerShell, un **programme modulaire**. Les scripts se basent sur un **ensemble de règles** (aussi appelé preset) que l'**administrateur peut activer ou non**. Chaque cas d'usage réel nous permet de mieux les classer. En effet, SWMB est à l'heure actuelle riche de plus de 500 règles. Chacune d'elle permet d'activer une fonction et dispose de son « anté-règle » qui permet de la désactiver. Cela laisse plus de 250 réglages possibles…*

*Afin d'appliquer un **set de règles de sécurité sur son parc Windows 10**, il suffit de l'exécuter en mode batch selon une **planification appropriée** : lors de l'installation de la machine et/ou au démarrage et/ou à l'ouverture de session… En vue de simplifier les tâches d'administration, un installeur de type setup.exe et des paquetages d'installation automatisée pour différents systèmes sont proposés (WAPT, OCS Inventory, PDQ Deploy...).*

*Le **projet** qui se veut **modulaire** laisse **libre** l'administrateur de choisir facilement, selon son environnement, les **règles** qu'il souhaite **appliquer**. Cependant, le projet vise également à proposer des **presets de base** prédéfinis qui soient à la fois **conformes** à la **politique de sécurité** de l'ESR tout en étant **fonctionnels en production** sur les parcs informatiques. **SWMB** a aussi **vocation** à être **collaboratif** en collationnant et en capitalisant les idées de tous les participants. Le projet peut être approché à plusieurs niveaux : du mode débutant simplement en exécutant l'installeur ou en mode expert en créant ses presets et fonctions associées, en le couplant à ses propres outils, en adaptant ses planifications d'exécution…*

Mots Clefs

ANSSI, Windows 10, BitLocker, Cortana, Fuite d'information, GPO, PowerShell

1 Origine du projet

Et un jour, tout a commencé, non pas à partir de rien, mais d'un substrat d'idées, de concepts, de bouts de code disséminés dans le *cyberespace* ! On peut notamment citer les éléments suivants comme détonateurs.

– Une inquiétude des ASR pour le respect de la vie privée et de la confidentialité que ce soit dans les premières versions de Microsoft Windows 10, ou dans les mises à jour successives (« upgrades de build »).

– Les recommandations de l'ANSSI [1] (annexe A) sous forme de document PDF (avec captures d'écran) – « *Préoccupations relatives au respect de la vie privée et à la confidentialité des données sous Windows 10* » [2].

– La formation SIARSv2 [3] rejouée à la délégation Alpes et animée par David GRAS et Gabriel MOREAU lors de laquelle un certain nombre de stagiaires – souvent démunis sur le sujet – ont initié le débat.

– L'implication du RSSI de la DR11 pour la mise en parallèle de l'application des règles de sécurité et confidentialité selon l'environnement : Active Directory (GPO – DR11) et sans AD (Scripts – LEGI).

– Atchoum, oups Labstumm, un ensemble de scripts PowerShell déployés via OCS Inventory [4] au laboratoire LEGI pour que les machines ne « s'enrhument » pas trop.

– Une page web, régulièrement mise à jour, qui présente une comparaison détaillée des outils de sécurisation et de confidentialité pour Windows 10 [5].

À l'automne 2019, un noyau dur initial grenoblois, a désiré fédérer ces initiatives autour d'un projet collectif. Ces personnes, qui se connaissaient et échangeaient déjà sur d'autres sujets, pouvaient se rencontrer aisément et commencer à écrire du code concret, fonctionnel même balbutiant mais qui plaçait l'initiative dans la réalité. Une demande a été faite pour la création d'un Groupe de Travail (GT [6]) axé sur ce sujet, lors du comité d'animation annuel de RESINFO [7] en novembre 2019. Le nom SWMB [8] est arrivé à ce moment-là, tout naturellement. Le GT a réellement décollé avec

des réunions régulières lors du premier confinement de mars 2020. Après la mise en place d'un premier prototype fonctionnel, le GT s'est rapidement ouvert et élargi à toute la communauté ESR (Enseignement Supérieur et Recherche). Nous étions passés du concept à une première réalisation.

2 Objectifs du programme

SWMB fait beaucoup de choses, mais il ne fait pas le café. Voici quelques-uns de ses principaux objectifs.

– Fournir un ensemble de scripts, essentiellement écrits en langage PowerShell [9], lisibles, adaptables et (ré)utilisables dans des environnements hétérogènes.

– Être facile à enrichir, en écrivant ses propres fonctions, *preset* voire modules.

– Être compatible avec toutes plateformes de déploiement (OCS Inventory [4], WAPT [10], PDQ Deploy [11], etc.)

– Pouvoir être utilisé en environnement AD (Active Directory) ou « Groupe de Travail » (*i.e.* sans AD).

– S'affranchir de la contrainte de la gestion graphique des GPO (Stratégies de groupe – *gpedit.msc*) [12] et pouvoir ainsi historiser ses actions dans le temps.

– Pouvoir faire dans la mesure du possible des actions et leurs inverses (*Enable / Disable*).

– Donner la possibilité à chaque site d'appliquer un ensemble plus ou moins large de règles, selon le contexte et les besoins, en gardant une logique de souplesse et de modularité. S'il est toujours possible de se limiter à la configuration par défaut, chaque site reste maître en dernier ressort de sa propre politique de sécurité.

– Intégrer les modifications régulières de l'OS, liées au cycle de vie de Windows 10 (Mises à jour de *builds* semestrielles).

À noter cependant que pour d'évidentes raisons de ressources humaines, SWMB se focalise uniquement sur les dernières versions de Windows et s'affranchit volontairement des anciennes versions (« *builds* ») plus maintenues.

3 Organisation du groupe

Un groupe de travail ne peut vivre sans une animation soutenue et régulière. Voici rapidement comment nous avons essayé de fonctionner.

– Constitution d'un groupe de travail dans le cadre du réseau métier RESINFO, afin d'avoir une formalisation et un label du projet. Nous avons ainsi rendu régulièrement des comptes au comité de pilotage en précisant notre degré d'avancement, nos atouts et contraintes.

– Visioconférences organisées régulièrement sur la plateforme Rendez-Vous (RENATER [13]) pour confronter les idées, partager les connaissances, faire des démonstrations, des tests, dans les environnements de chacun. Hors période de vacances scolaires, une périodicité de 15 jours a été retenue, afin de laisser à chacun le temps de travailler ses objectifs.

– Liste de diffusion hébergée par RESINFO / Chat de l'IN2P3 [14] via l'authentification eduGAIN [15].

– Site web RESINFO [7].

– Groupe « de base » composé au départ de quatre personnes. L'appel à candidature et participation a été lancée via la liste nationale ASR. Au moment de la rédaction de ce document, il y a 22 inscrits sur la liste de diffusion dédiée et une petite dizaine de personnes participent plus activement.

– Avancement et suivi sur la forge Gitlab de l'IN2P3 [16] dédiée au projet SWMB [8]. Plusieurs sous-projets existent, dont le principal contient le code, mais aussi les tickets pour les éventuels bogues et/ou souhaits d'amélioration. D'autres projets concernent plus le pilotage du GT, ou par exemple, la rédaction d'articles dédiés…

– Comptes-rendus et suivi de chaque réunion sur l'Etherpad de l'IN2P3 [17]. Ces *pads* sont ensuite sauvegardés sous forme Markdown sur la forge Gitlab. Pour des raisons de liberté d'expression lors des réunions, l'URL des *pads* n'est pas en accès libre.

Dans leur ensemble, les membres du GT tiennent à grandement remercier RENATER, le GDS Mathrice (hébergeur des services de RESINFO) et le CC IN2P3 pour accueillir l'ensemble de nos activités. Sans eux, le GT SWMB ne pourrait vous présenter le projet à un stade aussi avancé de fonctionnement.

4 Installation et utilisation

4.1 Vocabulaire

Pour mieux comprendre SWMB, il est nécessaire de bien définir le vocabulaire utilisé. Le programme se divise en trois concepts principaux.

– Les ***tweaks*** sont des règles de base dans SWMB. En général, chaque *tweak* a son pendant. L'un fait, l'autre défait (*Enable / Disable* par exemple).

– Les ***presets*** sont des fichiers regroupant en leur sein un ensemble de *tweaks*. SWMB propose ainsi plusieurs jeux de *preset*, ceux-ci sont régulièrement mis à jour par la communauté.

– Les **modules** sont les implémentations des *tweaks* en PowerShell. Chaque module regroupe en général le code source de plusieurs *tweaks*, classés par grande catégorie.

Le code SWMB importe les modules « à chaud » avant de traiter les *tweaks* définis dans les *presets* un par un. Le fonctionnement interne de SWMB sera détaillé dans le paragraphe 5 : Le cœur des algorithmes.

4.2 Installation graphique

L'installeur peut être téléchargé depuis la page web du projet [18]. Aucun prérequis n'est nécessaire sauf avoir les droits d'administrateur lors de l'installation. L'installeur se charge de désinstaller la version précédente si elle existe (figure 1).

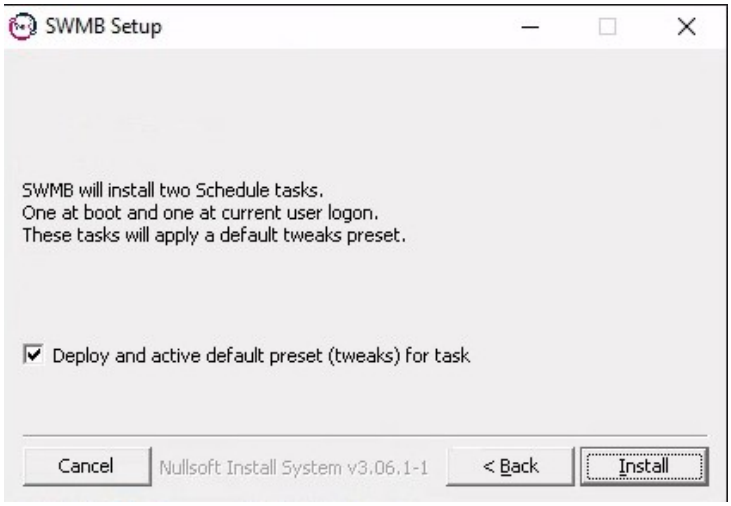

Figure 1: Installateur de SWMB

4.3 Installation en ligne de commande

```
SWMB-Setup-XXX.XXX.XXX.exe /S /ACTIVATED_PRESET=0
```

« XXX-XXX-XXX » est le numéro de version.

L'option */S* permet de réaliser une installation silencieuse (sans interface graphique).

L'option */ACTIVATED_PRESET=0* permet de ne pas déployer les fichiers de *preset* par défaut dans *C:\ProgramData\SWMB\Presets* (par défaut *ACTIVATED_PRESET=1*).

4.4 Les tâches planifiées

L'installation de SWMB (via l'installeur) ajoute deux tâches planifiées : *SWMB-LocalMachine-Boot* (au démarrage de la machine) et *SWMB-CurrentUser-Logon* (à l'ouverture de session d'un utilisateur). L'idée est d'appliquer ces règles après chaque redémarrage pour conserver le poste dans l'état voulu. Ainsi, quelles que soient les modifications apportées par un utilisateur, un programme ou une mise à jour Windows, la configuration souhaitée par l'administrateur du site est maintenue. Ces deux tâches exécutent un script différent, dont les fichiers de *presets* sont stockés dans le dossier *C:\ProgramData\SWMB\Presets*.

SWMB-LocalMachine-Boot exécute le script ***LocalMachine-Boot.ps1*** qui charge les modules contenant les fonctions liées à la configuration de la machine (clefs de registre HKLM). Le fichier de *preset* par défaut est *C:\ProgramData\SWMB\Presets\LocalMachine-Boot.preset.*

SWMB-CurrentUser-Logon exécute le script ***CurrentUser-Logon.ps1,*** qui charge les modules contenant les fonctions liées à la configuration de l'utilisateur (clefs de registre HKCU). Le fichier de *preset* par défaut est *C:\ProgramData\SWMB\Presets\CurrentUser-Logon.preset.*

4.5 Utilisation en ligne de commande

```
.\swmb.ps1 [option] TweakXX TweakYY…
```

Dans cet exemple ci-dessus, *TweakXX* et *TweakYY* sont les noms des règles à appliquer. Cela pourrait être *DisableWindowsStore* ou *EnableBitlocker*… Il y a un vaste choix possible ! Le programme *swmb.ps1* accepte en complément les options suivantes :

– ***-core*** doit être la première option si elle est utilisée. Elle importe seulement le module de base (minimal) *SWMB.psm1*, et non l'ensemble des autres modules déclarés dans *SWMB.psd1*.

– ***-import module_file.psm1*** importe le module susnommé dans SWMB. Vous pouvez étendre SWMB, tel quel, avec vos propres modifications. Cette option peut être déclarée autant de fois que nécessaire.

– ***-preset preset_file.preset*** charge tous les *tweaks* définis dans le fichier *preset*. Cette option peut être déclarée autant de fois que nécessaire.

– ***-log log_file*** *écrit tous* les messages dans le fichier de log et non dans le terminal.

– ***-check*** n'exécute pas les *tweaks* mais vérifie seulement leur existence (en accord avec le fichier prédéfini).

– ***-print*** n'exécute pas les *tweaks* mais en imprime la liste finale.

– ***-version*** affiche la version de SWMB.

– ***-exp*** est juste un raccourci pour importer le module *Modules\SWMB\Experimental.psm1*. Cette option est principalement utilisée par les développeurs pour aider à tester de nouvelles modifications.

– ***-hash hash_file.hash*** fait un *hash* SHA256 de la liste des *tweaks* (*preset*) et le compare avec l'ancien *hash* stocké dans le fichier passé en paramètre. Si les hachages diffèrent, un point de contrôle du système est effectué. Il est conseillé de placer le fichier de hachage dans le dossier *C:\ProgramData\SWMB\Caches* avec le nom du *preset* le plus important suivi de l'extension *.hash*.

Les *tweaks* peuvent être positionnés dans la ligne de commande. Un *tweak* commençant par un point d'exclamation (!) retire ce *tweak* de la liste des *tweaks* à traiter (contenus dans un *preset*), si celui-ci avait été activé précédemment.

Exemples – Déactiver la télémétrie :

```
.\swmb.ps1 DisableTelemetry
```

Passage d'un jeu de *preset* :

```
.\swmb.ps1 -preset "c:\Program Files\SWMB\Presets\LocalMachine-Default.preset"
```

Jeu de *preset* sans le *tweak DisableTelemetry* :

```
& 'C:\Program Files\SWMB\swmb.ps1' `
  -preset "c:\Program Files\SWMB\Presets\LocalMachine-Default.preset" !DisableTelemetry `
  -log c:\SwmbLog.log
```

Il est aussi possible de forcer l'exécution d'une tâche programmée. Par exemple en PowerShell :

```
Start-ScheduledTask -TaskName "SWMB-LocalMachine-Boot"
```

4.6 Fichiers de log

Les tâches planifiées écrivent un fichier de journal dans *C:\ProgramData\SWMB\Logs*. Les fichiers se nomment *CurrentUser-lastLogon.log* pour la tâche planifiée de démarrage de session et *LocalMachine-LastBoot.log* pour la tâche programmée de *boot*. Ces fichiers contiennent le résultat de chacune des fonctions paramétrées dans le fichier de *preset*. Ils permettent de contrôler le bon déroulement de l'exécution de ces fonctions.

Par ailleurs, SWMB écrit dans l'observateur d'événements de Windows (*eventvwr.msc*) dans « *Journaux Windows > Application > source SWMB* ». Nous avons des messages sur le démarrage des tâches programmées, par exemple : « *SWMB: Run Logon Script for User username – Start* ». Voir aussi l'option *-log* quand le script est lancé manuellement.

4.7 Utilisation graphique

SWMB dispose d'une interface graphique minimale qui permet de réaliser quelques actions (figure 2).

Figure 2: Interface graphique minimale

– Lancer le script interactif de chiffrement des lecteurs avec BitLocker (voir § 6.1.6).

– Suspendre ou Reprendre BitLocker.

– Exécuter immédiatement la tâche programmée de démarrage de la machine.

– Indiquer la présence d'une mise à jour disponible de SWMB.

Cette interface est en cours de développement et devrait s'enrichir de fonctionnalités dans les versions à venir.

5 Le cœur des algorithmes

5.1 Utilisation d'un projet existant

Après une étude comparative de différentes solutions (voir [5]), nous avons fait le choix de suivre un projet plutôt que de « réinventer la roue et partir from scratch ». Celui de Disassembler0 nous semblait le plus proche de nos attentes [19]. SWMB en était très proche au démarrage (*fork*, puis synchronisation successive). Il est désormais complètement indépendant, notamment de part l'archivage du projet amont. Le code de Disassembler0 avait les qualités suivantes :

– très clair et bien écrit ;

– une routine principale réduite à sa plus simple expression ;

– modulaire et facilement extensible via l'import de code en ligne de commande ;

– toute action avait sa contre-action. Le code permettait de faire et de défaire de la même manière ;

– les actions étaient regroupées sous forme de fichier de *preset*, chargeable en ligne de commande ;

– de très nombreuses actions étaient déjà définies.

Ce projet était la meilleure base de départ que nous ayons analysé. Cependant, son développement avait été réalisé avec un tout petit nombre de fichiers, sans import de modules au lancement, sans procédure standardisée d'installation, etc. Il nous a donc fallu adapter le code pour le rendre plus simple à étendre, à développer en collaboratif et *in fine* à utiliser.

5.2 Arborescence du programme

Un minimum de fichiers sont à la racine du projet, principalement *swmb.ps1* le code principal et *wisemoui.ps1* la couche graphique minimale. Les fichiers importants sont répartis dans les dossiers suivants.

5.2.1 Modules

Ce dossier regroupe le module principal *SWMB.psm1* qui intègre les routines du cœur des algorithmes et le module *SWMB.psd1* qui permet de charger tous les modules secondaires, placés et classés par nom dans le sous-dossier *Modules\SWMB*. Il y a dans ce dossier deux catégories de modules, ceux qui concernent l'ordinateur en tant que tel (*LocalMachine*) et ceux qui concernent l'utilisateur courant (*CurrentUser*). Dans chaque module sont définis des fonctions, permettant de faire ou défaire des règles de comportement (sécurité au sens large)

- *Modules/SWMB/CurrentUser-Application.psm1*
- *Modules/SWMB/CurrentUser-Privacy.psm1*
- *Modules/SWMB/LocalMachine-Network.psm1*
- *Modules/SWMB/LocalMachine-Privacy.psm1*
- *Modules/SWMB/LocalMachine-Security.psm1*
- …

5.2.2 Presets

Ce dossier des fichiers de *presets* est mis à disposition de la communauté. Avec plus de 500 *tweaks*, il n'est pas toujours facile de savoir lesquelles prendre ! De la même manière, les fichiers de *presets* sont soit pour la machine, soit pour l'utilisateur courant. Les fichiers *Presets\LocalMachine-All.preset* et *Presets\LocalMachine-All.preset* regroupent l'ensemble des *tweaks*. Pour faciliter le classement et la recherche, les *tweaks* sont regroupés en catégorie dont les noms évoquent clairement le domaine d'application :

- *Privacy Tweaks*
- *UWP Privacy*
- *Security Tweaks*
- *Network Tweaks*
- *Service Tweaks*
- *UI Tweaks*

- *Explorer UI Tweaks*
- *Application Tweaks*
- *Server Specific Tweaks*
- *BitLocker Tweaks*

À noter que le GT propose deux fichiers de *preset* marqués « *Recommended* ». Il s'agit de deux listes relativement stables, qui ont été testées sur plusieurs sites et permettent de respecter au mieux les recommandations de l'ANSSI sans être bloquantes. Il est possible de définir ses propres listes qui incluent celles-ci.

5.2.3 Tasks

Deux tâches planifiées ont été définies lors de l'installation, une lors du démarrage de la machine (*Tasks\LocalMachine-Boot.ps1)* et l'autre lors de l'ouverture de la session utilisateur (*Tasks\ CurrentUser-Logon.ps1)*. Ces deux tâches sont par défaut actives.

Une troisième tâche permet de lancer le chiffrement de la machine avec BitLocker. C'est un exemple un peu différent mais qui est utile, car il permet d'analyser comment utiliser SWMB dans un cas particulier.

5.2.4 Setup

Ce dossier contient des scripts de post-installation et de pré-suppression utile à l'installeur. Celui-ci est défini dans le fichier *package.nsis* au format NSIS à la racine du projet.

5.2.5 Dists

Il y a plusieurs exemples d'utilisation ou de déploiement de SWMB dans cette arborescence. Il y a par exemple, un sous dossier pour le système OCS Inventory et un autre pour WAPT. Les fichiers ici présents ne sont pas déployés sur une machine via l'installeur. En annexe C, nous montrons comment déployer SWMB avec PDQ Deploy.

5.2.6 ProgramData

Afin de ne pas modifier le dossier d'installation de SWMB, les fichiers créés à l'exécution ou les paramètres sont à positionner dans le dossier *C:\ProgramData\SWMB*. Il y a actuellement quatre sous-dossiers qui suivent à peu près la même architecture que le dossier d'installation : *Presets*, *Modules*, *Logs* et *Caches*. Ces deux derniers permettent d'y enregistrer des données temporaires comme des traces, ou des résultats de fonction de hachage comme nous le verrons plus loin.

5.3 Fonctionnement de la boucle principale, lecture des modules

Le programme principal est divisé en deux parties. La première lit les paramètres de la ligne de commande. Il s'agit d'une cascade assez classique de tests. Puis, un ensemble de modules sont dynamiquement chargés et certaines fonctions exécutées. L'algorithme, très simple, est décrit ci-dessous. On peut noter que le fonctionnement très général du code, qui n'a pas de fonction réellement spécialisée sur la sécurité dans son noyau, permet d'envisager l'utilisation de SWMB pour d'autres aspects que la sécurité.

```
# Loading the SWMB base engine with all the main modules (neested)
Import-Module Modules\SWMB.psd1
# Initialize
SWMB_Init
# Loop on module (option -import)
Import-Module Modules\XXXX.psm1
# Loop on preset file (option -preset)
```

```
# Each preset file is a suite of tweaks
SWMB_LoadTweakFile "Presets\YYYY.preset"
# Load one tweak (can be called multiple times)
# Unloads the tweak if it starts with the exclamation mark (!)
SWMB_AddOrRemoveTweak "ZZZZ"
# Execute all loaded tweaks (presets)
SWMB_RunTweaks
```

Le module principal est chargé, puis c'est au tour des modules optionnels « *XXXX* » déclarés en ligne de commandes. Entre-temps, l'initialisation de SWMB remet à zéro certaines valeurs comme la table de tous les *tweaks* à traiter. Une fois cette étape de chargement de code finie, les fichiers de *presets* « *YYYY* » sont lus. Les *tweaks* sont enregistrés dans une table et s'il y a des inclusions, soit de modules (*$IMPORT*), soit de *preset* (*$PRESET*), ceux-ci sont traités de manières récursives. Il est possible de rajouter ou de supprimer (avec !) un *tweak* « *ZZZZ* » dans la table globale directement depuis la ligne de commande ou un fichier de *preset*.

La dernière étape consiste alors à exécuter tous les tweaks dans leur ordre d'apparition.

6 Cas particuliers

6.1 Chiffrement BitLocker

SWMB permet de chiffrer un poste Windows avec BitLocker [20] selon un cahier des charges bien précis. La fonction est disponible sous forme d'un script interactif. De cette manière, tous les postes disposent des mêmes paramètres de chiffrement, ce qui garantit l'homogénéité du parc.

6.1.1 Cahier des charges

Nous avons établi une liste de conditions et d'objectifs pour un chiffrement homogène et sécurisé.

– Ordinateur configuré en mode de démarrage UEFI avec Secure Boot [21] ;

– Puce TPM dans l'état « prête » [22] ;

– Utilisation de l'algorithme de chiffrement *XtsAes256* [23] ;

– Proposition de chiffrement de tous les disques internes et déchiffrement automatique des disques non système ;

– Création automatique des fichiers contenant les clefs de chiffrement, avec la possibilité de sauvegarder ces clefs sur un lecteur réseau ;

– Possibilité de mettre un code PIN ;

– Prise en compte de l'état initial du poste (ordinateur déjà chiffré, chiffrement en cours, etc.).

6.1.2 Exécution

```
.\swmb.ps1 EnableBitlocker
```

Il est possible d'activer le chiffrement en suivant les recommandations du CNRS en ligne de commande ou depuis l'interface graphique (voir § 4.7).

6.1.3 Algorithme

L'algorithme de décision suit le schéma de la figure 3. En fonction de différents tests, nous proposons une action à l'utilisateur : chiffrement, redémarrage, changement de méthode de chiffrement ou attente.

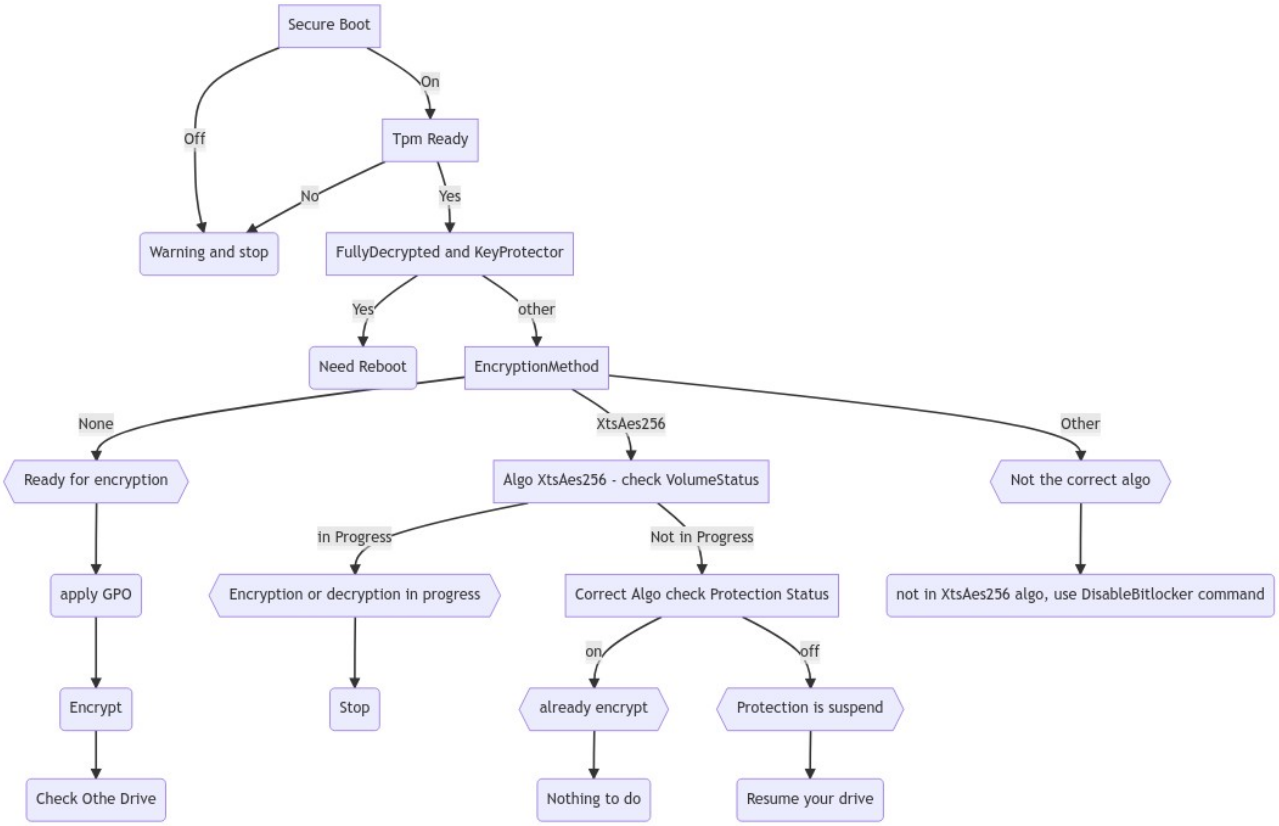

Figure 3: Algorithme de chiffrement BitLocker

Avant le chiffrement, nous configurons BitLocker afin de fixer quelques paramètres de sécurité en modifiants les clefs de registres suivantes : *HKLM:\SOFTWARE\Policies\Microsoft\FVE* [24]:

– pour forcer l'utilisation de l'algorithme de chiffrement *XtsAes256*, la clef *EncryptionMethodWithXtsOs* prend la valeur 7 ;

– pour empêcher l'utilisateur de modifier le code PIN, la clef *DisallowStandardUserPINReset* prend la valeur 1.

6.1.4 Clefs de chiffrement / déchiffrement

Par défaut, les fichiers de clefs sont stockés sur le disque système. Le script propose aussi de les stocker sur un espace réseau. Des droits particuliers sont appliqués sur les fichiers contenant ces clefs. Ces fichiers ne peuvent être lus par aucun utilisateur (même administrateur). Elles peuvent être simplement copiées par un compte administrateur.

Il appartient ensuite à l'administrateur du parc de sauvegarder ces fichiers de clefs dans un espace sécurisé et de les effacer du disque système.

6.1.5 Déchiffrement

Il existe une fonction de déchiffrement qui décode tous les lecteurs non-amovibles. Attention, n'étant plus chiffrés, cette opération sur les disques doit de préférence rester temporaire.

```
.\swmb.ps1 DisableBitlocker
```

6.1.6 Interface graphique

L'interface graphique de SWMB propose quelques fonctions autour de BitLocker (voir § 4.7) :

– *Suspend / Resume* ;

– Chiffrer tous les disques en lançant le script dans une console dédiée.

6.2 SWMB & Active Directory

Le programme SWMB peut être déployé via les GPO d'un domaine AD, au moyen d'un script de démarrage (pour la partie ordinateur) ou à l'ouverture de session (pour la partie utilisateur).

Actuellement, SWMB n'est pas disponible au format *.msi* (Microsoft Software Installer), format spécifique conçu pour l'installation, la mise à jour et la désinstallation des logiciels, nativement pris en charge en environnement AD. Il n'est donc pas (encore) possible d'utiliser les GPO « *Paramètres du logiciel* > *Installation de logiciel* » prévues à cet effet, car elles ne sont utilisables qu'avec des packages *msi (*Figure 4 – AD avec MSI).

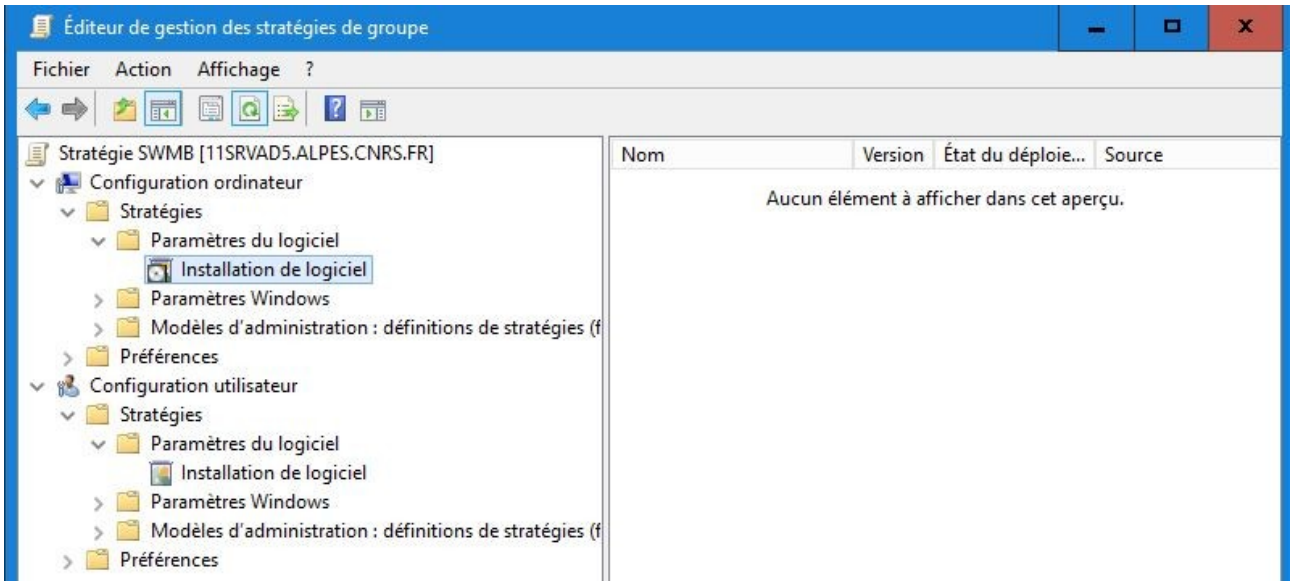

Figure 4: AD avec msi

En revanche, le programme exécutable (*Swmb-Setup.exe*) peut être lancé comme décrit par le site it-connect [25]. Il faut alors utiliser les GPO « *Paramètres Windows* > *Scripts* » et ajouter un script PowerShell qui lancera l'exécutable au démarrage de la machine ou à l'ouverture de session de l'utilisateur (figure 5). Il est également possible de le déployer ponctuellement à l'installation d'un nouveau poste, comme n'importe quelle autre application, en utilisant les outils et solutions telles que WDS / MDT / SCCM…

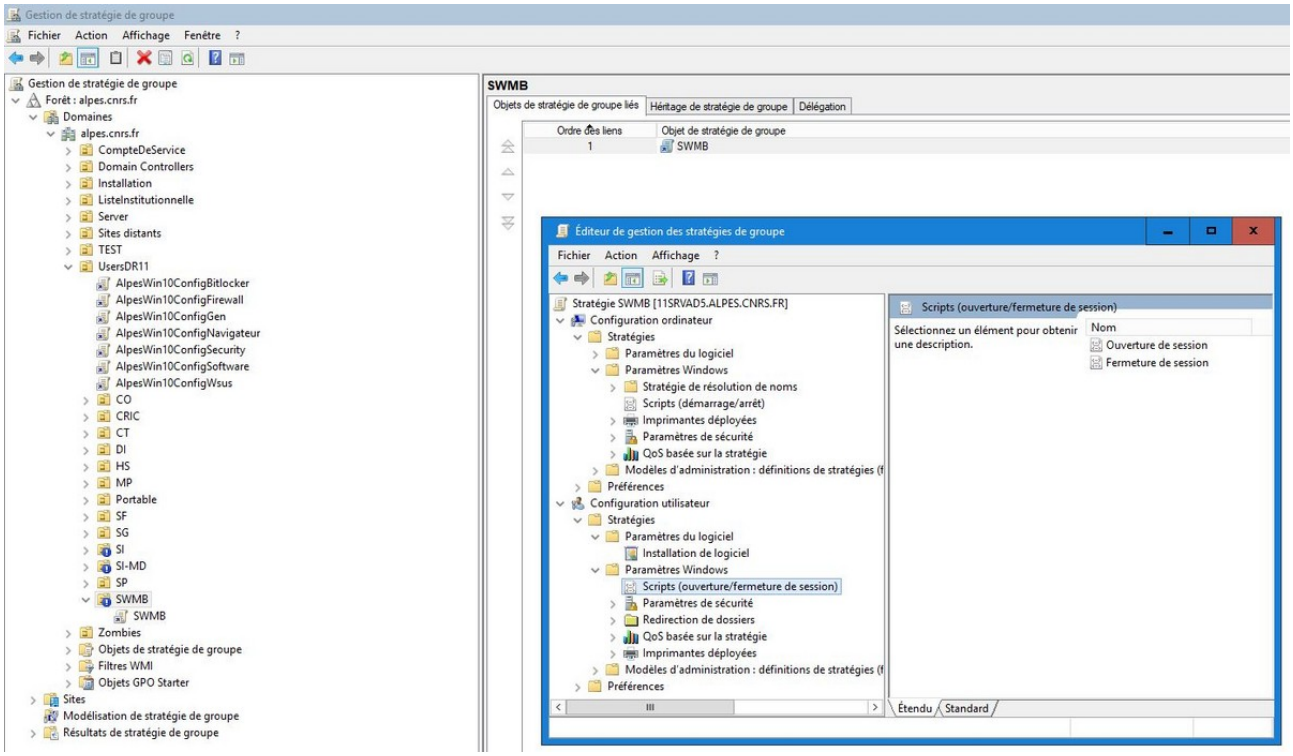

Figure 5: AD déploiement avec script

Au premier abord, proposer l'utilisation de SWMB dans le contexte AD peut sembler étrange, car les GPO systèmes produits par Microsoft (ou par d'autres), disponibles nativement, peuvent faire double emploi. Il est cependant envisageable de « mixer » les deux, en réservant le programme SWMB aux réglages généraux, propres à l'ensemble du parc, et dédier les GPO intégrées aux configurations plus spécifiques.

Cela implique en revanche d'être rigoureux et clair sur « quel outil fait quoi ? ».

Pour éviter des conflits ou l'application en doublon de paramètres, Il convient de définir clairement cette répartition : les réglages système « figés » ayant peu ou pas de raison d'être modifiés peuvent être appliqués avec SWMB une fois pour toutes. Cela permet d'éviter d'avoir un nombre trop important de GPO activées, ce qui peut rendre fastidieuse leur gestion et utilisation au sein d'un AD. On réserve ainsi les GPO déployées via l'AD aux configurations plus fines des applications, en touchant le moins possible au système.

De plus l'application par SWMB permet de s'affranchir de la « dépendance » des GPO au domaine. Par exemple une machine en mode « nomade » qui n'est pas connectée au VPN et donc isolée de l'AD peut se voir forcer l'application de réglages. Exemple : désactivation du compte d'utilisateur, après un certain temps sans connexion au domaine. Il est aussi à noter que très souvent dans un parc, quelques machines peuvent être hors domaine, comme le contrôle d'accès, un ordinateur de pilotage…

Certains jeux de *preset* SWMB permettent de gérer des paramètres non prévus ou configurables en GPO, par exemple, la suppression de la plupart des applications liées au Microsoft Store, qui n'ont pas lieu d'être installées en contexte professionnel.

6.3 Failles de sécurité

Les failles de sécurité – particulièrement les *0-day* – sont un point délicat mais aussi un challenge pour le GT, car elles nécessitent une réactivité importante. Ces informations de vulnérabilité arrivent par différents canaux : chaîne fonctionnelle sécurité, liste de diffusion métier, veille technologique. Le groupe essaie de coller à ces informations, mais si vous êtes informés, vous pouvez commencer votre implication dans le projet en créant une « *new issue* » sur le dépôt Git du projet. Et mieux, si vous avez la solution de contournement, proposez là (voir annexe E) !

Il est à noter que dans certains cas, le palliatif est simple (telle l'ajout, la suppression ou la modification d'une clef de registre) et la correction pourra être rapidement intégrée (création du *tweak* de la fonction associée et intégration dans un *preset*). Mais dans d'autres cas, plus de temps d'analyse et de développement sont nécessaires. C'est pourquoi, de nouveau, si vous avez la solution, n'hésitez pas à proposer votre solution pour intégration ou intégrez vous-même le groupe !

Prenons l'exemple de la faille CVE-2021-40444 [26] datant de début septembre concernant l'exécution à distance de code dans le composant MSHTML. Le GT, au vu de la documentation trouvée sur internet, a rapidement mis en place un *tweak* assez simple à programmer permettant un contournement (voir § 7.2),

```
# Disable
Function TweakDisableMSHTMLActiveX { # RESINFO
	Write-Output "Disable ActiveX in MSHTML (CVE-2021-40444)..."
	For ($zone = 0 ; $zone -le 3 ; $zone++){
		If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\$zone")) {
			New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\$zone" -Force | Out-Null
		}
		Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\$zone" -Name "1001" -Type DWord -Value 00000003
		Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\$zone" -Name "1004" -Type DWord -Value 00000003
	}
}
```

ainsi que son *tweak* inverse.

```
# Enable
Function TweakEnableMSHTMLActiveX { # RESINFO
	Write-Output "Enable ActiveX in MSHTML (CVE-2021-40444)..."
	For ($zone = 0 ; $zone -le 3 ; $zone++){
		Remove-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\$zone" -Name "1001" -ErrorAction SilentlyContinue
		Remove-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\$zone" -Name "1004" -ErrorAction SilentlyContinue
	}
}
```

Il se pose également la question de la manière de créer votre *tweak*, de mettre à jour et déployer les fichiers de *preset* de votre site, afin de tenir compte de ce contournement. Si cela n'est pas réalisé rapidement par le GT dans la branche principale du projet, le paragraphe 7.1 donne quelques pistes pour le faire par vous-même.

À l'inverse, lorsque l'éditeur déploie un patch de sécurité corrigeant la faille, la question du maintien du contournement se pose, comme toujours avec ce type de solution palliative. Selon les cas, il sera possible soit de maintenir le contournement, soit de l'annuler avant la mise en place du

patch éditeur ou après (mais il y a le risque d'annuler des modifications du patch lui-même). Les différentes possibilités sont plus ou moins contraignantes et intrusives dans la gestion d'un parc machine.

Ce sont des questions que le groupe se pose mais pour lesquelles les réponses n'ont pas encore été tranchées !

7 Utilisations avancées

7.1 Ajouts de règles dans les fichiers de *presets* afin de déployer son propre jeux de règles.

Par défaut deux fichiers de *presets* (correspondant à la configuration machine et utilisateur) sont copiés avec quelques règles recommandées et validées par le groupe RESINFO. Ils sont automatiquement mis à jour avec chaque nouvelle version de SWMB, car ils contiennent la chaîne magique « file automatically updated ». Si cette chaîne de caractères est absente, ils ne seront pas mis à jour. Cela permet à chaque site d'avoir ses propres fichiers, sans craindre qu'une mise à jour de SWMB ne vienne écraser la configuration spécifique du site.

De plus, lors de l'installation, il est possible de ne pas utiliser ces fichiers prédéfinis par défaut, en décochant une case dans le programme d'installation (ou avec l'option */ACTIVATED_PRESET=0* en ligne de commande).

Exemple d'un fichier de *preset C:\ProgramData\SWMB\Presets\CurrentUser-Logon.preset :*

```
# Utiliser un module personnalisé
$IMPORT "C:\MyFonctions.psm1"
# Utiliser un tweak de notre module
DisableMyTweak
# Utiliser un preset recommandé par Resinfo
$PRESET "C:\Program Files\SWMB\Presets\CurrentUser-Logon-Recommanded.preset"
# Sauf le tweak suivant
!ShowKnownExtensions_CU
```

Le déploiement du fichier de *preset* peut se faire avec la méthode de votre choix. Par exemple, avec un script PowerShell.

```
$CurrentUserlogonFile =  "$Env:ProgramData\SWMB\Presets\CurrentUser-Logon.preset"
#méthode pour écrire dans un fichier
Set-Content -Path "$CurrentUserlogonFile" -Value '#MyLABO preset'
Add-Content -Path "$CurrentUserlogonFile" -Value "SysEvent"
Add-Content -Path "$CurrentUserlogonFile" -Value "`$PRESET `"$Env:ProgramFiles\SWMB\Presets\
CurrentUser-All.preset\`""
Add-Content -Path "$CurrentUserlogonFile" -Value "SysEvent"
#méthode pour écrire directement un nouveau fichier
Rename-Item -Path "$CurrentUserlogonFile" -NewName ("$CurrentUserlogonFile"+".old") -Force -
ErrorAction Ignore
Copy-Item -force "$PSScriptRoot\CurrentUser-Logon-Legi.preset" -destination "$CurrentUserlogonFile"
```

7.2 Méthode pour ajouter une règle

La configuration de Windows consiste bien souvent à utiliser une interface graphique (par exemple « *gpedit.msc* » ou la fenêtre de « *Paramètres* » du système) pour modifier une fonctionnalité de Windows. Cette interface entraîne principalement la modification d'une ou plusieurs clefs de registre de Windows. L'écriture de règles avec SWMB se résume donc souvent à la modification de ces clefs de registre, en écrivant quelques lignes de PowerShell.

En écrivant cette règle sous la forme d'une fonction et en rédigeant par symétrie une anté-fonction qui réalise l'opération inverse (c'est facultatif, mais plus propre pour une intégration future dans le projet), il est facile de créer son propre module qui pourra être utilisé au sein de SWMB.

Nous disposons de plusieurs outils pour retrouver les clefs modifiées via une interface graphique.

7.2.1 Sites internet de GPO

Ces sites internet proposent une vue similaire à celle de « *gpedit* ». Ils sont très complets et permettent de rechercher via de nombreux filtres.

– Group Policy Search [27]

– Group Policy Administrative Templates Catalog [28]

– Microsoft propose un tableur *Excel* qui contient toutes les GPO des systèmes d'exploitation officiellement supportés [29].

7.2.2 Utilisation de Process Monitor

C'est le moyen le plus sûr de contrôler les clefs de registres modifiées par une interface graphique. *Process Monitor* est un des utilitaires de la suite *SysInternals* de Microsoft [30].

Nous allons illustrer la méthode à partir d'un exemple provenant du document de l'ANSSI « sécurisation de Windows 10 » [2].

L'ANSSI recommande dans son annexe A.4 : « *GPO de désactivation de l'envoi de données par l'antivirus Windows Defender* » de mettre la valeur « *désactivée* » à la GPO « *Configurer une valeur de remplacement de paramètre locale pour l'envoi de rapports à Microsoft MAPS* ».

Notre trouvons cette GPO depuis *gpedit* dans « *Configuration ordinateur / Modèles d'administration / Composants Windows / Antivirus Windows Defender / MAPS / Configurer une valeur de remplacement de paramètre locale pour l'envoi de rapports à Microsoft MAPS* »

Pour comprendre quelle clef est modifiée, il suffit de suivre la procédure suivante :

1. Ouvrir *gpedit* et *Process Monitor* (et rien d'autre) ;
2. Ajouter les deux filtres suivants dans process monitor : « *Process Name* » is « mmc.exe » qui correspond à la console que l'on vient d'ouvrir et « *Operation* » is « *RegSetValue* » qui est une action d'écriture d'une clé de registre (figures 6 et 7) ;

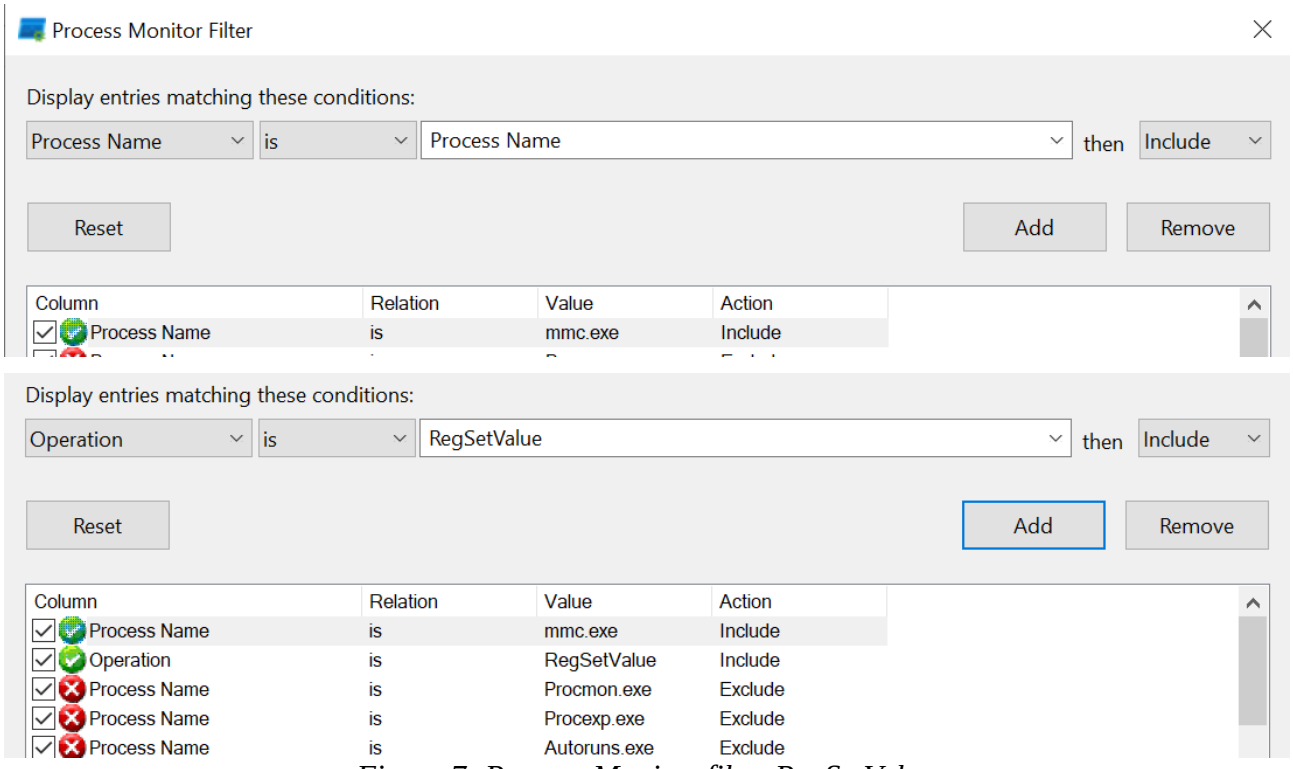

Figure 7: Process Monitor filtre RegSetValue

3. Modifier la clef avec *gpedit* ;
4. Lire les valeurs modifiées dans *Process Monitor (figure 8*).

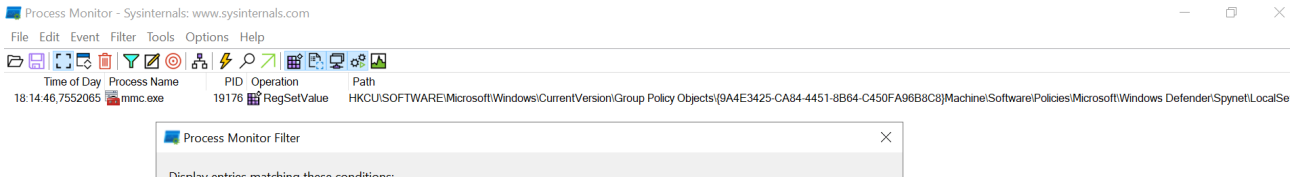

Figure 8: Process Monitor - capture du registre modifiée

Dans notre cas, nous obtenons sur *Process Monitor* :

Ordinateur\HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy Objects\{9A4E3425-CA84-4451-8B64-C450FA96B8C8}Machine\Software\Policies\Microsoft\Windows Defender\Spynet\LocalSettingOverrideSpynetReporting => data 0 ou data 1 en fonction de la valeur « Activé » ou « Désactivé ».

Nous pouvons aussi retrouver la clef de registre depuis le site web ADMX [31].

L'écriture en PowerShell de la règle est :

```
Function TweakDisableOverrideReportingMAPS { # RESINFO
    Write-Output "Disabling override for reporting to Microsoft MAPS…"
    If (!(Test-Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet")) {
        New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet" -Force | Out-Null
    }
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet" -Name "LocalSettingOverrideSpynetReporting" -Type DWord -Value 0}


# Enable
Function TweakEnableOverrideReportingMAPS { # RESINFO
    Write-Output "Enabling override for reporting to Microsoft MAPS…"
    Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet" -Name "LocalSettingOverrideSpynetReporting" -Type DWord -Value 1 -ErrorAction SilentlyContinue
}
```

8 Limitations

Bien que chaque action ait son inverse (ou anté-action) dans 95 % des cas, quelques-unes ne permettent pas un « retour arrière » à l'identique. Seule une sauvegarde ou la création d'un point de restauration le permet. À noter que cela n'est pas spécifique à SWMB, les autres outils fonctionnent de la même manière. En effet, SWMB fonctionne sans état. Lorsqu'un *tweak* est appliqué, il n'est pas noté (tracé) s'il a une action sur le système, ou si celui-ci était déjà dans cet état. L'état initial d'un poste informatique n'est donc pas connu. C'est principalement l'état final qui nous intéresse. Ce principe rend lui aussi plus complexe l'annulation d'une action.

En pratique, il n'y a quasiment aucune différence entre un *tweak* et une GPO, les deux modifiant juste des clefs de registres ici ou là dans la majorité des cas. Cependant, le système de GPO intégré dans Windows maintient sa propre base de données et ne tient pas compte de l'état réel du poste. Ainsi l'application des *tweaks* ne se retrouve malheureusement pas dans la console d'état des stratégies de groupe. Il est ainsi possible, si on n'y prête pas garde, d'appliquer des stratégies incompatibles, le poste de travail basculant alors régulièrement d'un état à un autre.

Cependant, le projet est jeune ce qui a permis des évolutions régulières dans l'architecture interne. Les premiers utilisateurs ont dû s'adapter, car le développement suit les usages avec cette approche Agile du développement. Des pistes existent pour combler ces limitations et apporter d'autres améliorations dans l'usage du produit. Mais pour cela, il faut des bras et du temps (voir annexe E) !

9 Retour d'expérience au laboratoire LEGI

Le laboratoire LEGI (UMR5519) utilise SWMB sur un parc d'environ 80 machines Windows 10 depuis novembre 2020. Il est déployé au travers d'un script maison (Labstumm) avec OCS Inventory [4] dont un agent tourne sur chaque machine du laboratoire.

Dans un premier temps, le script d'application des règles de SWMB était uniquement exécuté à l'installation de la machine et à chaque nouveau déploiement de notre système de configuration (environ une fois tous les deux mois). Depuis le début de l'automne 2021, tous les nouveaux postes de travail, portables et fixes ont les tâches planifiées activées. Au vu du bon fonctionnement de ses postes, cette fonctionnalité va être déployée à l'ensemble du parc très prochainement.

Les fichiers de configuration du laboratoire sont positionnés en annexe D, expurgé des quelques points sensibles qui n'ont pas forcément vocation à être rendu publiques. Souvent, une bonne chose à faire est de commencer par charger la configuration préconisée par RESINFO. Ainsi, nous profitons des nouvelles avancées du projet collaboratif. Autre point important, notre propre bibliothèque est chargée dès le début comme le montre les quelques lignes ci-dessous. Ainsi, nos propres paramétrages peuvent être mis petit à petit en forme sous forme de *tweak*, et lorsqu'ils sont suffisamment bien écrits et s'ils sont relativement génériques, peuvent être proposés au GT pour une intégration dans SWMB (annexe E).

```
# Use local module
$IMPORT "C:\Program Files\LabStumm\LegiFonctions.psm1"
# Use Resinfo recommanded preset
$PRESET "C:\Program Files\SWMB\Presets\LocalMachine-Boot-Recommanded.preset"
```

Au cours des mois passés, le script d'installation *maison* s'est ainsi fortement réduit en longueur. Les différentes parties, réécrites sous forme de fonction dans un module à part, permettent un fonctionnement plus standard profitant de toutes les fonctionnalités des tâches planifiées. Par exemple, notre propre gestion de l'ouverture de session a ainsi été supprimé puisque SWMB le réalise parfaitement, et c'est ainsi que les copieurs couleurs basculent régulièrement pour les utilisateurs en mode noir et blanc par défaut afin de limiter les coûts.

SWMB a permis au laboratoire LEGI de progresser encore dans la qualité de configuration des postes Windows 10. Tout est plus propre et plus lisible. C'est mieux documenté avec un fonctionnement plus clair, tout en conservant l'historique de toutes les versions qui était une culture déjà bien ancrée dans notre service informatique.

10 Conclusion et perspectives

Deux ans après sa création, le projet SWMB est passé du statut de concept au statut de projet fonctionnel et opérationnel, bénéficiant d'un installateur graphique. Initialement utilisé dans un laboratoire comme une sous-partie d'un script d'installation, il est devenu le script principal s'exécutant à chaque démarrage et à chaque ouverture de session. De part son aspect modulaire, il a été possible et facile de faire cette inversion des rôles. Les postes sont désormais plus sécurisés avec des *presets* appliqués plus régulièrement et un traçage plus fin des actions réalisées, permettant ainsi de détecter plus rapidement des bogues de configuration en cas d'erreurs.

Les recommandations de l'ANSSI [1] sont transcrites sous forme d'action et de contre-action. Nous avons commencé à intégrer les recommandations du BSI [32] (annexe B). De nouveaux *tweaks* sont aussi venus s'ajouter au cours des discussions et des alertes de sécurité. Ainsi, une partie interactive est venue se greffer, pour simplifier la configuration de BitLocker sur les postes, en restant cependant dans l'architecture modulaire de SWMB. Il est aussi possible de paramétrer certains *tweaks* comme la configuration des serveurs de temps. Au final, beaucoup de choses sont déjà possibles, mais beaucoup restent à faire.

Chaque site doit rester maître de sa politique. En proposant, en n'imposant pas, SWMB est dans cette stratégie de mutualisation constructive. Il peut dépasser le concept de la sécurité pour se rapprocher de la configuration générale. Par exemple, un site peut décider, entre autres, de sa stratégie d'impression en fonction de ses photocopieurs. Ce site peut ainsi déployer ses propres règles et SWMB les appliquera au même titre que les autres, lors des tâches planifiées. Ce scénario est déjà opérationnel sur un site en production. À noter qu'avoir l'historique de tous les codes, *presets* compris, dans une forge Git permet de savoir exactement qui a proposé quoi et à quelle date. La sécurité des postes de travail sous Windows est ainsi versionnée.

Dans le futur, l'intégration de Windows 11 ne posera pas de problème, ce nouvel OS étant une simple évolution du système d'exploitation Windows 10. Nous pouvons cependant donner quelques orientations : simplifier et compléter la documentation permettant d'utiliser et d'étendre SWMB, simplifier encore l'ajout de fonctionnalités via un système de greffons, ajouter des règles de sécurité aux *presets* proposés par défaut, proposer régulièrement quelques séances de formation, tester, tester et encore tester…

Nous souhaitons que de nombreux sites s'approprient SWMB (annexe E), l'adoptent et l'adaptent pour leur site, puis que ces expériences enrichissent SWMB lui-même, afin que la sécurité d'un parc machine ne soit plus l'affaire d'un site, mais qu'elle soit mutualisée par tous les sites.

SWMB a besoin de vous !

11 Licence

Bibliographie

[1] ANSSI – Agence Nationale de la Sécurité des Systèmes d'Information : https://www.ssi.gouv.fr/

[2] Recommandations de l'ANSSI – « *Préoccupations relatives au respect de la vie privée et à la confidentialité des données sous Windows 10* » : https://www.ssi.gouv.fr/uploads/2017/01/np_securisation_windows10_collecte_de_donnees_v1.2.pdf

[3] Formation SIARSv2 : https://resinfo.org/les-groupes-de-travail-11/groupe-de-travail-siars/article/siars-v2

[4] OCS Inventory : https://ocsinventory-ng.org/

[5] Martin Brinkmann – Comparaison des outils de sécurisation et de confidentialité pour Windows 10 : https://www.ghacks.net/2015/08/14/comparison-of-windows-10-privacy-tools/

[6] Groupe de Travail SWMB : https://resinfo.org/les-groupes-de-travail-11/groupe-de-travail-swmb/

[7] RESINFO – Fédération des réseaux métiers d'**A**dministrateurs **S**ystèmes et **R**éseaux dans l'Enseignement et la Recherche : https://resinfo.org/

[8] SWMB : https://gitlab.in2p3.fr/resinfo-gt/swmb/resinfo-swmb

[9] PowerShell : https://docs.microsoft.com/fr-fr/powershell/

[10] WAPT : https://fr.wikipedia.org/wiki/Wapt_(logiciel)

[11] PDQ Deploy : https://www.pdq.com/

[12] GPO / Stratégie de groupe : https://fr.wikipedia.org/wiki/Stratégie_de_groupe

[13] RENATER : https://www.renater.fr/

[14] CC-IN2P3 : https://cc.in2p3.fr/

[15] eduGAIN : https://edugain.org/

[16] Forge Gitlab de l'IN2P3 : https://gitlab.in2p3.fr/

[17] Etherpad de l'IN2P3 : https://etherpad.in2p3.fr/

[18] Installateur de SWMB : https://resinfo-gt.pages.in2p3.fr/swmb/resinfo-swmb/

[19] Win10-initial-Setup-Script : https://github.com/Disassembler0/Win10-Initial-Setup-Script

[20] BitLocker : https://fr.wikipedia.org/wiki/BitLocker_Drive_Encryption

[21] UEFI – *Unified Extensible Firmware Interface* : https://fr.wikipedia.org/wiki/UEFI

[22] TPM – *Trusted Platform Module* : https://fr.wikipedia.org/wiki/Trusted_Platform_Module

[23] BitLocker settings reference for Windows 10 and later : https://docs.microsoft.com/en-us/mem/configmgr/protect/tech-ref/bitlocker/settings#windows-10-or-later-devices

[24] Choose drive encryption method and cipher strength : https://admx.help/?Category=MDOP&Policy=Microsoft.Policies.BitLockerManagement::BLEncryptionMethodWithXts_Name

[25] Florian Burnel – site it-connect : https://www.it-connect.fr/comment-deployer-un-logiciel-au-format-exe-par-gpo/

[26] Vulnérabilité d'exécution de code à distance dans Microsoft MSHTML : https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-40444

[27] Group Policy Search : https://gpsearch.azurewebsites.net/

[28] Group Policy Administrative Templates Catalog / ADMX : https://admx.help/

[29] Group Policy Settings Reference for Windows : https://www.microsoft.com/en-us/download/details.aspx?id=25250

[30] Procmon : https://docs.microsoft.com/en-us/sysinternals/downloads/procmon

[31] Envoi de rapports à Microsoft MAPS : https://admx.help/?Category=Windows_10_2016&Policy=Microsoft.Policies.WindowsDefender::Spynet_LocalSettingOverrideSpynetReporting&Language=fr-fr

[32] BSI – Bundesamt für Sicherheit in der Informationstechnik : https://fr.wikipedia.org/wiki/Office_f%C3%A9d%C3%A9ral_de_la_s%C3%A9curit%C3%A9_des_technologies_de_l%27information – https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Cyber-Security/SiSyPHuS/AP11/Hardening_Guideline.pdf

[33] Licence Art Libre : https://artlibre.org/

[34] Tickets ou issues SWMB : https://gitlab.in2p3.fr/resinfo-gt/swmb/resinfo-swmb/-/issues

[35] Liste de diffusion SWMB : https://listes.resinfo.org/wws/info/swmb-gt

[36] Chat du GT SWMB : https://chat.in2p3.fr/group/resinfo-gt-swmb

[37] Guide pour contribuer à SWMB : https://gitlab.in2p3.fr/resinfo-gt/swmb/resinfo-swmb/-/blob/master/CONTRIBUTING.md

A – Annexe ANSSI

L'**Agence Nationale de la Sécurité des Systèmes d'Information** (ANSSI) est un service français créé en 2009. Elle est rattachée au Secrétariat Général de la Défense et de la Sécurité Nationale (SGDSN), autorité chargée d'assister le Premier ministre dans l'exercice de ses responsabilités en matière de défense et de sécurité nationale. Elle remplace la Direction Centrale de la Sécurité des Systèmes d'Information (DCSSI), créée en 2001.

Le document « Guide Windows 10 – ANSSI » date de près de cinq ans, posant les grandes lignes des règles de confidentialité préconisées pour Windows 10. Cependant, compte tenu des nombreuses évolutions du système depuis 2017, il n'est pas exhaustif, la dernière version prenant en compte la *build* 1703 de Windows 10 (Remarque : 9 « *builds* » parues depuis, à raison d'une tous les 6 mois environ. Dernière build en date : 21H2 – Octobre 2021).

Principaux points traités dans le document : collecte des données personnelles et télémétrie, assistant Cortana (et Windows Search), personnalisation expérience utilisateur, applications Microsoft intégrées (liées à MS Store), cloud, (compte Microsoft, One Drive).

Le document se veut adaptable au contexte d'utilisation et au niveau de confidentialité souhaité pour les données. L'ANSSI définit quatre niveaux de recommandation, selon le besoin de protection / confidentialité : général (quel que soit le contexte), standard, élevé et très élevé. Le niveau de collecte des données par le service de télémétrie illustre bien ces quatre niveaux : sécurité (pas de télémétrie – Uniquement versions entreprise et éducation) de base, amélioré et complet.

Les outils utilisés pour mettre en œuvre les recommandations sont de différentes natures : GPO utilisateur ou ordinateur, appliquées au démarrage ou à la connexion, scripts PowerShell.

B – Annexe BSI

L'**Office fédéral de la sécurité des technologies de l'information** (*Bundesamt für Sicherheit in der Informationstechnik* ou BSI) est une administration allemande créée en 1991 et chargée de la sécurité des technologies de l'information et de la communication. Il s'occupe notamment de la sécurité des logiciels, de la protection des infrastructures de communications, de la sécurité dans le cyberespace, de cryptographie, de contre-écoute électronique, de certification de produits de sécurité et de l'accréditation de laboratoires de test.

L’agence propose un document similaire à celui de l'ANSSI intitulé « *Configuration Recommendations for Hardening of Windows 10 Using Built-in Functionalities* » qui a été réalisé par une société nommée ERNW GmbH pour le compte du Federal Office (BSI).

Ils définissent 3 niveaux de « durcissement » qui couvrent les principaux usages :

1 « *Normal protection needs domain member* » – Ce premier profil vise la protection contre des attaques non ciblées et des infections type malwares, et ne restreignent pas les fonctionnalités du système ;

2 « *Increased protection needs domain member* » – Ce second profil vise la protection contre des attaques et des malwares ciblés. Des restrictions « acceptables » de fonctionnalités du système sont attendues ;

3 « N*ormal protection needs standalone computer* » – Ce troisième profil vise, comme le premier, la protection contre des attaques non ciblées et des infections type malwares, et ne restreignent pas les fonctionnalités du système, à la différence près qu’il s’applique à des systèmes hors domaine.

Les objectifs de leurs recommandations sont les suivants :

– Prévenir les attaques connues et largement diffusées et exploitées ;

– Réduire la surface d’attaque ;

– Améliorer la protection des données en désactivant les services « cloud » ;

– Améliorer la protection des données en désactivant toutes les communications vers Microsoft autant que possible.

Les règles intégrées sont celles qui peuvent être directement exprimées en PowerShell et couvrent les domaines suivants :

– Désactivation de la télémétrie Windows ;

– Désactivation des anciennes versions de PowerShell qui ne proposent pas de fonctionnalités de sécurité avancées ;

– Définition des politiques d’exécution de PowerShell (*Restricted*, *AllSigned*, *RemoteSigned*, ou *Unrestricted*), et applicables (*Scope*) à un utilisateur, à la session PowerShell en cours, ou à tous les utilisateurs d’une machine donnée ;

– Désactivation de l’utilisation de PowerShell à distance ;

– Désactivation de Windows Script Host en local et à distance.

C – Annexe déploiement de SWMB avec PDQ Deploy

Dans un souci de limiter les exemples, nous montrons ici comment SWMB est déployé avec PDQ Deploy [11]. Il faut cependant savoir qu’il est actuellement déployé sur des sites avec OCS Inventory [4] ou avec WAPT [10] de manière assez proche dans l’esprit.

PDQ Deploy est un logiciel commercial de télé-déploiement proposé par la société Admin Arsenal, conçu pour aider les organisations à installer des programmes, des mises à jour et des correctifs logiciels via un portail web unifié. Il est l’outil choisi par l’équipe de gestion de parc de la DSI de l’Université Grenoble Alpes, et est utilisé ici pour déployer le logiciel SWMB sur un parc Windows 10 Education UGA.

Pour cela, le dernier paquet proposé par le projet SWMB est téléchargeable depuis le site du projet [18]. L'installeur SWMB téléchargé est ensuite déposé dans le répertoire des logiciels du serveur PDQ, via un partage de fichiers SMB standard.

Un nouveau package est créé dans *PDQ Deploy* > *New Package*. Le nom, la version, et la description du package réalisé sont renseignés dans l'onglet *Details* (figure 9).

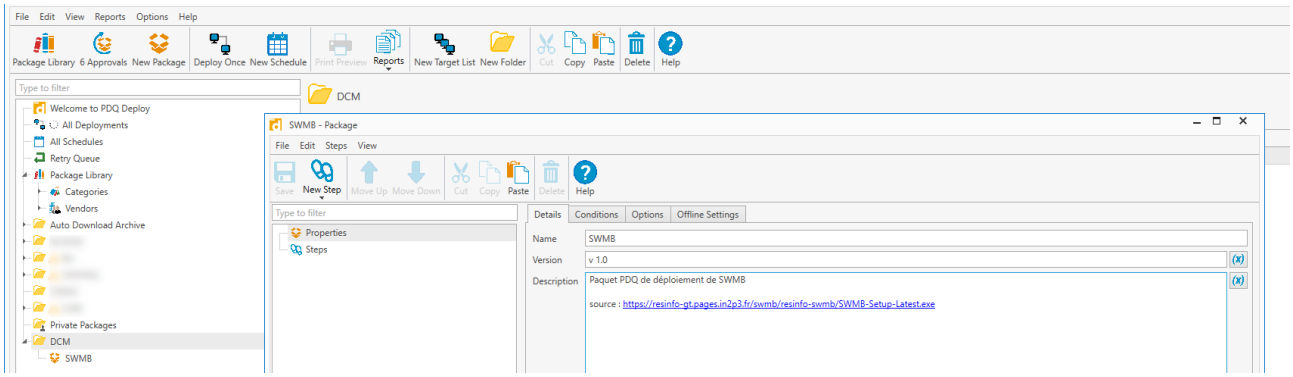

Figure 9: PDQ - new package

Dans l'onglet Conditions, les systèmes d'exploitation et leur version (32 ou 64 bits) sont définis, auxquels le package pourra s'appliquer (figure 10).

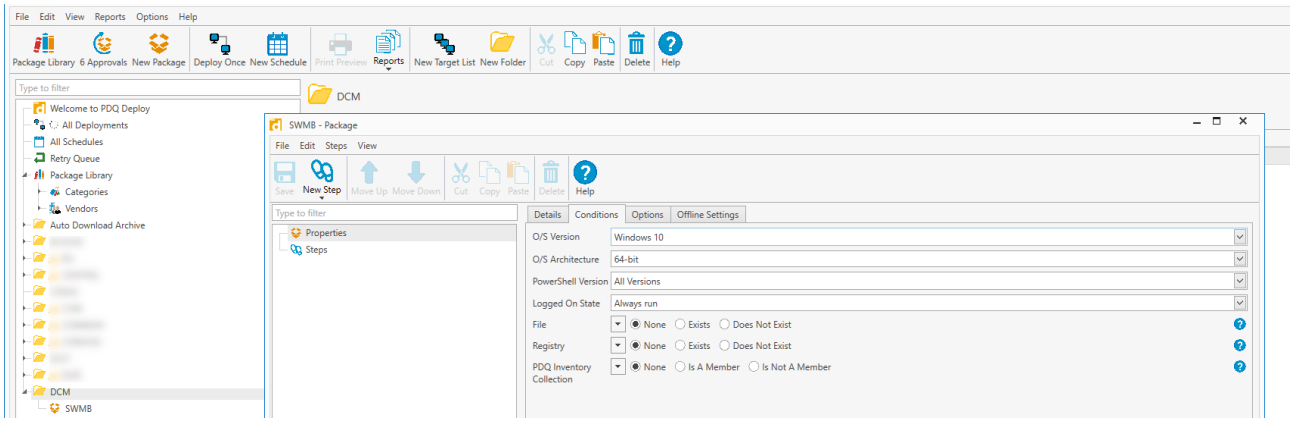

Figure 10: PDQ - new package conditions

Un 1er Step « *Install* » est choisi pour lancer l'exécutable, ici *SWMB-Setup-Latest.exe*, auquel est ajouté le paramètre */S* pour l'installation silencieuse (figure 11).

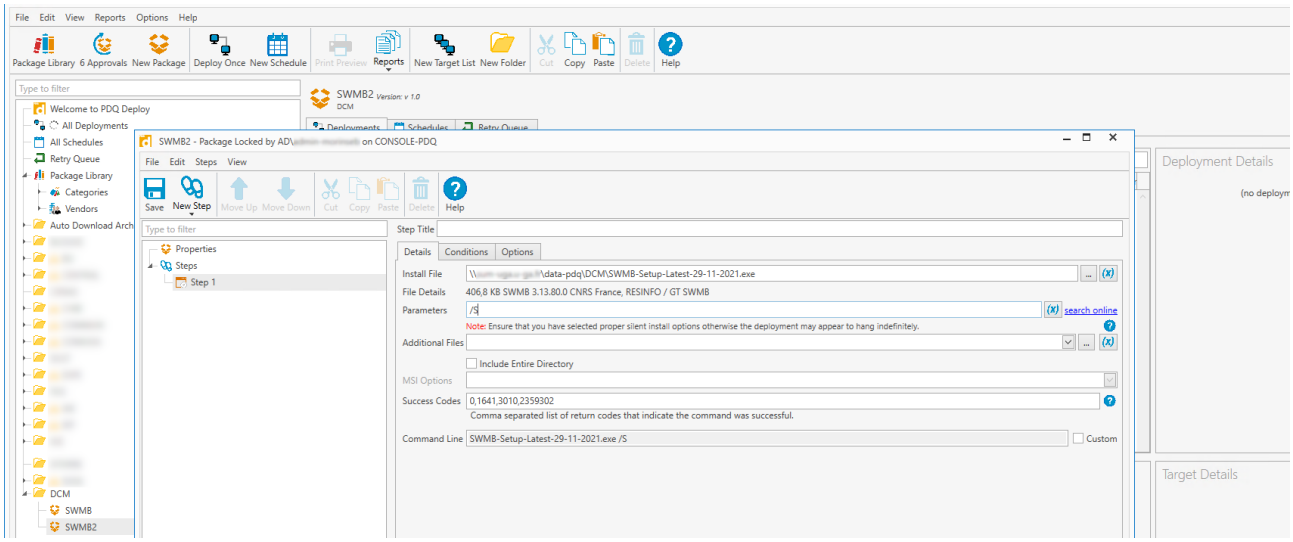

Figure 11: PDQ - choix de l'exécutable

Un 2ème Step « *Command* » est ajouté pour lancer un script de post-installation, qui lancera SWMB une toute première fois sur le poste client sur lequel le logiciel a été déployé, et proposera une trace de l'exécution du logiciel. Puis le package est définitivement validé avec le bouton « *Save* » (figure 12).

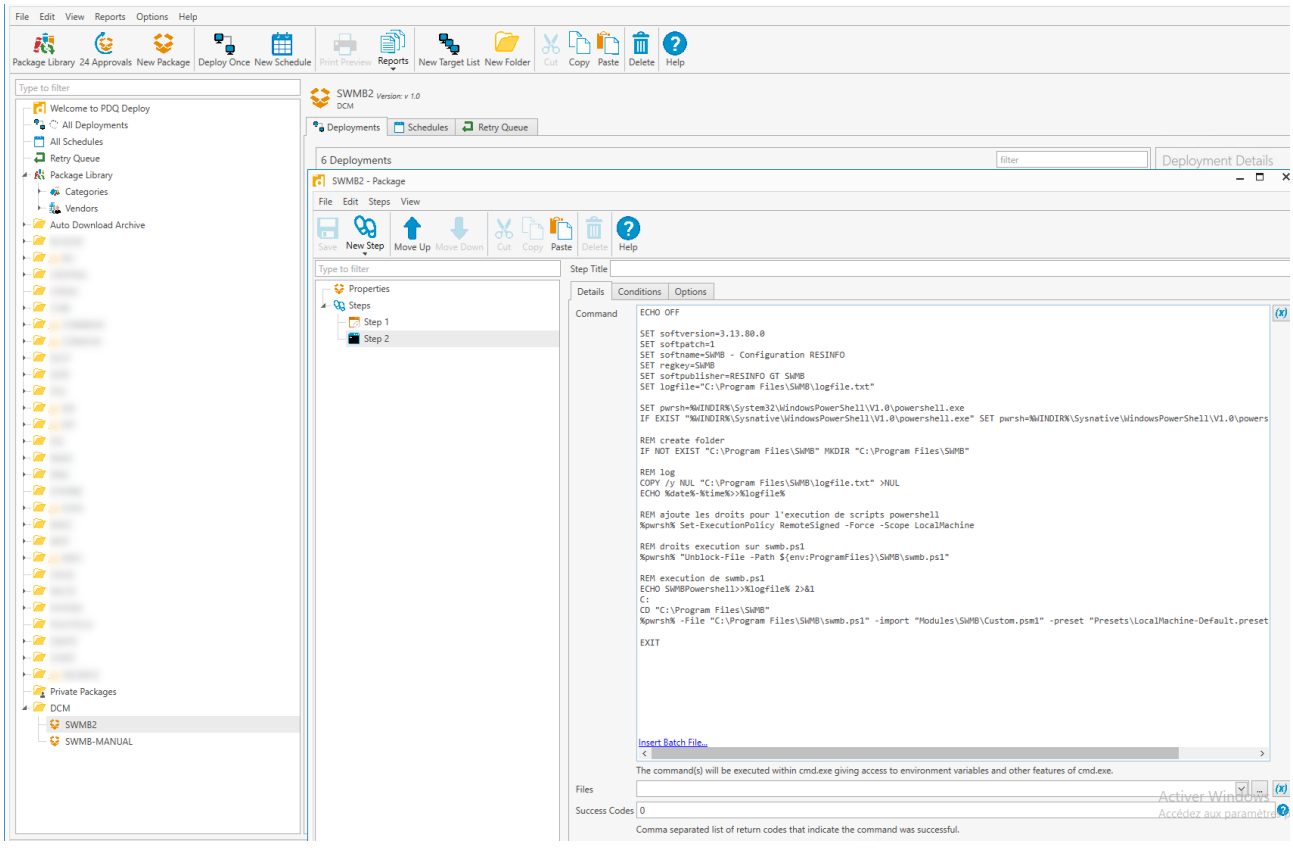

Figure 12: PDQ - script post-install

Le package est terminé et prêt à être déployé, Pour cela, sélectionner le package et d'un clic-droit, choisir « *Deploy Once* ». Le ou les ordinateurs destinataires du package sont définis avec « *Add computer* » et le déploiement démarre par un clic sur « *Deploy Now* » (figure 13).

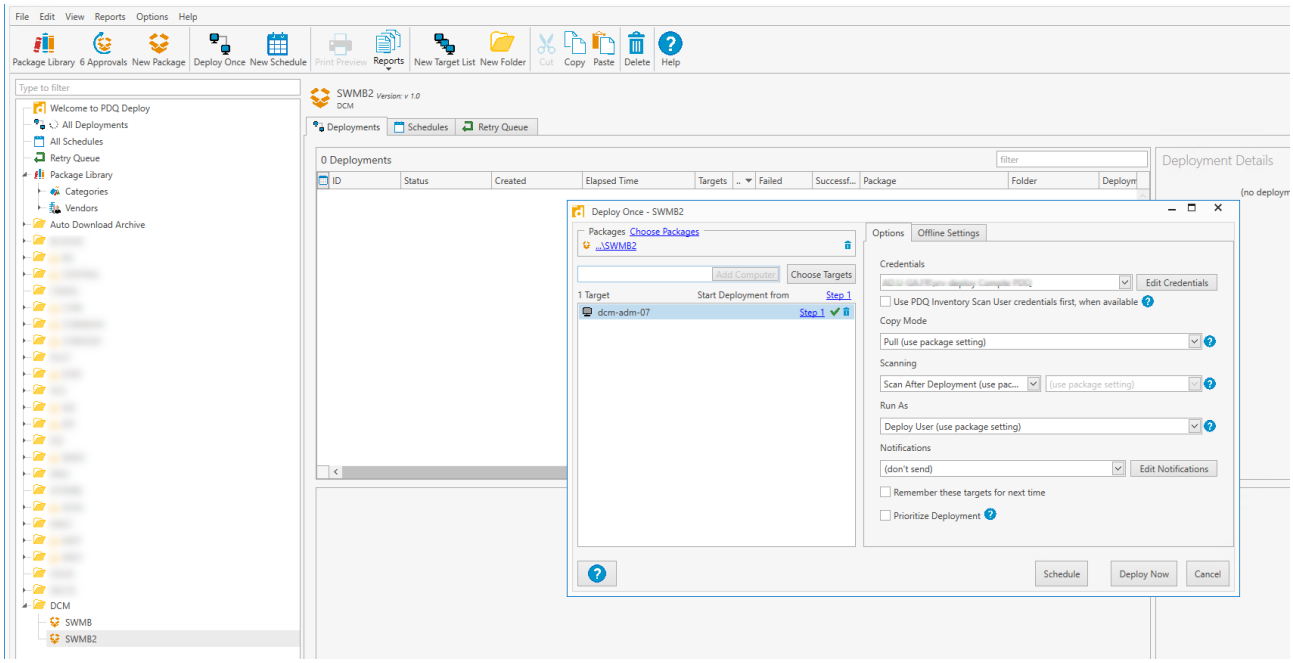

Figure 13: PDQ - choix des targets de destination

L'installation se lance, le serveur PDQ se connecte à l'ordinateur destinataire. L'installation se déroule automatiquement, PDQ copie tout d'abord le fichier sur l'ordinateur de destination et au bout de quelques dizaines de secondes, la copie, l'installation du package, et son déploiement sont confirmés par un *Successful* (figure 14).

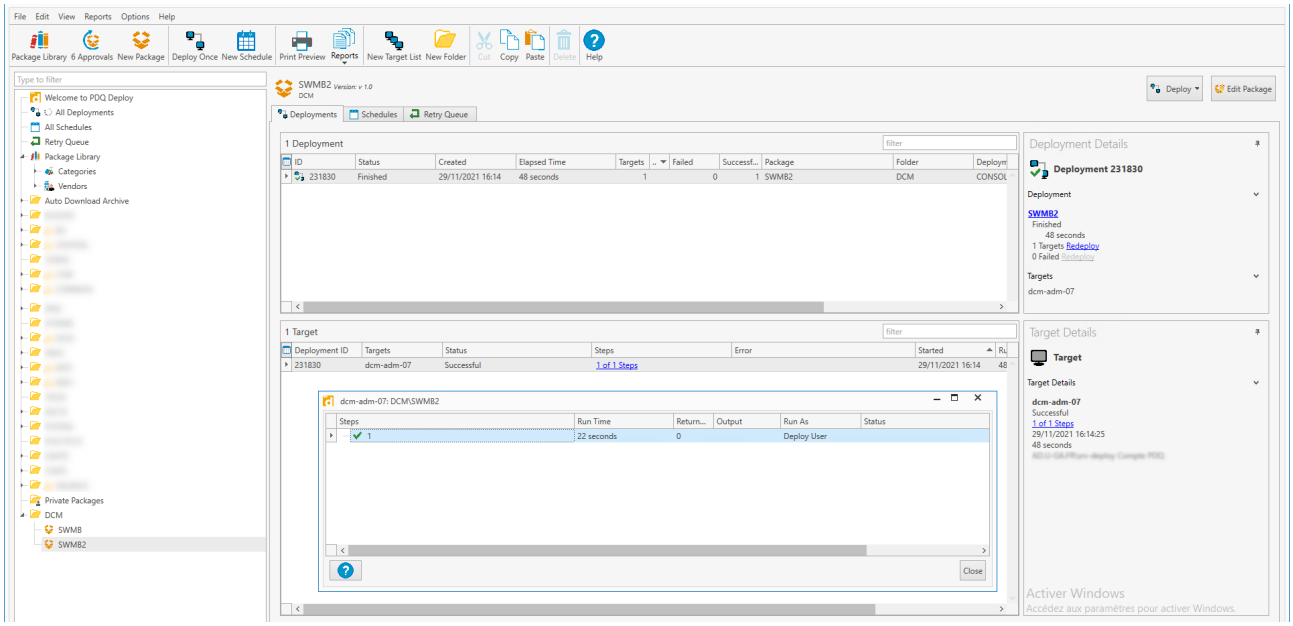

Figure 14: PDQ - déploiement succesful

Sur le poste client destination, les tâches planifiées pour lancer SWMB ont bien été ajoutées automatiquement (figure 15).

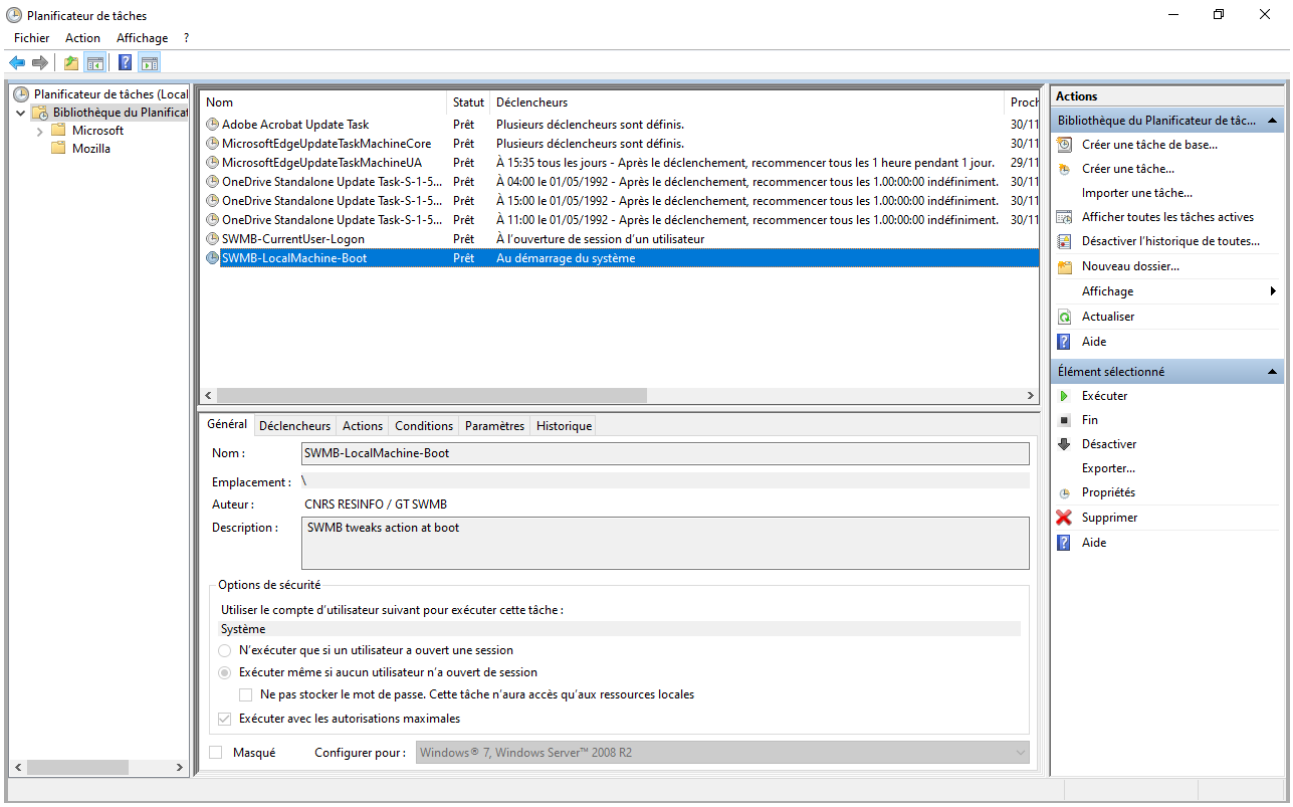

Figure 15: PDQ - tâche planifiée

Toujours sur le poste client destination, le programme SWMB apparaît désormais bien dans la liste des programmes installés.

Le programme SWMB est désormais installé sur le poste client de destination, le logiciel a été lancé une première fois juste après l'installation, et deux tâches planifiées se chargeront de ré-appliquer la configuration par défaut validée par les développeurs de SWMB, au boot, et au login, au cas où des mises à jour système auraient modifié certains des réglages souhaités.

D – Annexe règles de déploiement de SWMB au LEGI

Règles appliquées au démarrage de la machine

```
##########
# LEGI
# Tweak for LocalMachine
##########

### Require administrator privileges ###
SysRequireAdmin

# Use local module
$IMPORT "C:\Program Files\LabStumm\LegiFonctions.psm1"

# Use Resinfo recommanded preset
$PRESET "C:\Program Files\SWMB\Presets\LocalMachine-Boot-Recommanded.preset"

### Privacy Tweaks ###
DisableTelemetry                # EnableTelemetry
```

```
DisableCortana                  # EnableCortana
DisableWiFiSense                # EnableWiFiSense
DisableAppSuggestions           # EnableAppSuggestions
DisableActivityHistory          # EnableActivityHistory
DisableLocation                 # EnableLocation
DisableMapUpdates               # EnableMapUpdates
DisableFeedback                 # EnableFeedback
DisableAdvertisingID            # EnableAdvertisingID
DisableErrorReporting           # EnableErrorReporting
DisableDiagTrack                # EnableDiagTrack
DisableWAPPush                  # EnableWAPPush
DisablePrivacyExperience        # EnablePrivacyExperienc

### UWP Privacy Tweaks ###
DisableUWPBackgroundApps        # EnableUWPBackgroundApps
DisableUWPNotifications         # EnableUWPNotifications
DisableUWPAccountInfo           # EnableUWPAccountInfo
DisableUWPDiagInfo              # EnableUWPDiagInfo
DisableUWPAccessLocation        # EnableUWPAccessLocation

### Security Tweaks ###
DisableAdminShares              # EnableAdminShares
DisableScriptHost               # EnableScriptHost
EnableDotNetStrongCrypto        # DisableDotNetStrongCrypto
EnableF8BootMenu                # DisableF8BootMenu
SetDEPOptOut                    # SetDEPOptIn

### Network Tweaks ###
SetCurrentNetworkPrivate        # SetCurrentNetworkPublic
DisableRemoteAssistance         # EnableRemoteAssistance

### Service Tweaks ###
EnableUpdateMSProducts          # DisableUpdateMSProducts
DisableUpdateRestart            # EnableUpdateRestart
DisableMaintenanceWakeUp        # EnableMaintenanceWakeUp
DisableAutorun                  # EnableAutorun
EnableNTFSLongPaths             # DisableNTFSLongPaths

### UI Tweaks ###
DisableLockScreen               # EnableLockScreen
HideNetworkFromLockScreen       # ShowNetworkOnLockScreen
HideShutdownFromLockScreen      # ShowShutdownOnLockScreen
DisableLockScreenBlur           # EnableLockScreenBlur
DisableSearchAppInStore         # EnableSearchAppInStore
DisableNewAppPrompt             # EnableNewAppPrompt

### Explorer UI Tweaks ###
HideQuickAccess                 # ShowQuickAccess
HideDesktopFromThisPC           # ShowDesktopInThisPC
HideDocumentsFromThisPC         # ShowDocumentsInThisPC
HideDownloadsFromThisPC         # ShowDownloadsInThisPC
HideMusicFromThisPC             # ShowMusicInThisPC
HidePicturesFromThisPC          # ShowPicturesInThisPC
HideVideosFromThisPC            # ShowVideosInThisPC
```

```
Hide3DObjectsFromThisPC         # Show3DObjectsInThisPC

### Application Tweaks ###
DisableOneDrive                 # EnableOneDrive
UninstallOneDrive               # InstallOneDrive
UninstallMsftBloat              # InstallMsftBloat
UninstallThirdPartyBloat        # InstallThirdPartyBloat
DisableXboxFeatures             # EnableXboxFeatures
DisableAdobeFlash               # EnableAdobeFlash
DisableEdgePreload              # EnableEdgePreload
DisableEdgeShortcutCreation     # EnableEdgeShortcutCreation
DisableIEFirstRun               # EnableIEFirstRun
DisableFirstLogonAnimation      # EnableFirstLogonAnimation
DisableMediaSharing             # EnableMediaSharing

SetPhotoViewerAssociation       # UnsetPhotoViewerAssociation
AddPhotoViewerOpenWith          # RemovePhotoViewerOpenWith
UninstallXPSPrinter             # InstallXPSPrinter
RemoveFaxPrinter                # AddFaxPrinter

### Local Tweaks ###
ShowKnownExtensions
SetNTPConfig
InstallLocalPrinter
DisableDellSupport
```

Règles appliquées au démarrage de chaque session utilisateur

```
##########
# LEGI
# Tweak for CurrentUser
##########

# Use local module
$IMPORT "C:\Program Files\LabStumm\LegiFonctions.psm1"

# Use Resinfo recommanded preset
$PRESET "C:\Program Files\SWMB\Presets\CurrentUser-Logon-Recommanded.preset"

### Privacy Tweaks ###
DisableCortana_CU               # EnableCortana_CU
DisableTailoredExperiences_CU   # EnableTailoredExperiences_CU
DisableWebLangList_CU           # EnableWebLangList_CU
DisableAppSuggestions_CU        # EnableAppSuggestions_CU

### UWP Privacy Tweaks ###
DisableUWPBackgroundApps_CU     # EnableUWPBackgroundApps_CU

### Security Tweaks ###
HideAccountProtectionWarn_CU    # ShowAccountProtectionWarn_CU

### Service Tweaks ###
DisableSharedExperiences_CU     # EnableSharedExperiences_CU
DisableAutoplay_CU              # EnableAutoplay_CU
```

```
### UI Tweaks ###
DisableActionCenter_CU          # EnableActionCenter_CU
DisableAccessibilityKeys_CU     # EnableAccessibilityKeys_CU
ShowTaskManagerDetails_CU       # HideTaskManagerDetails_CU
ShowFileOperationsDetails_CU    # HideFileOperationsDetails_CU
HideTaskbarSearch_CU            # ShowTaskbarSearchIcon_CU      # ShowTaskbarSearchBox_CU
HideTaskView_CU                 # ShowTaskView_CU
ShowSmallTaskbarIcons_CU        # ShowLargeTaskbarIcons_CU
SetTaskbarCombineWhenFull_CU    # SetTaskbarCombineNever_CU     # SetTaskbarCombineAlways_CU
HideTaskbarPeopleIcon_CU        # ShowTaskbarPeopleIcon_CU
ShowTrayIcons_CU                # HideTrayIcons_CU
DisableShortcutInName_CU        # EnableShortcutInName_CU
SetVisualFXPerformance_CU       # SetVisualFXAppearance_CU
DisableF1HelpKey_CU             # EnableF1HelpKey_CU

### Explorer UI Tweaks ###
ShowKnownExtensions_CU          # HideKnownExtensions_CU
ShowHiddenFiles_CU              # HideHiddenFiles_CU
EnableNavPaneExpand_CU          # DisableNavPaneExpand_CU
HideSyncNotifications_CU        # ShowSyncNotifications_CU
HideRecentShortcuts_CU          # ShowRecentShortcuts_CU
SetExplorerThisPC_CU            # SetExplorerQuickAccess_CU
ShowThisPCOnDesktop_CU          # HideThisPCFromDesktop_CU
DisableThumbnailCache_CU        # EnableThumbnailCache_CU
DisableThumbsDBOnNetwork_CU     # EnableThumbsDBOnNetwork_CU

### Application Tweaks ###
DisableXboxFeatures_CU          # EnableXboxFeatures_CU

### Local Tweaks ###
SetPrintBW_CU
```

E – Annexe participation au projet

Contribuer au projet SWMB est accessible à tout le monde, le logiciel SWMB est proposé sous licence libre MIT. Si vous êtes curieux et désireux de rejoindre le groupe de travail [6], il y a toujours des choses que vous pouvez faire, selon le temps et les compétences dont vous disposez**.**

Tests & QA – Il est préférable de distribuer un logiciel orienté sécurité en étant sûrs de son bon fonctionnement ! **Testez le logiciel SWMB sur votre parc** (logiciel, paquetages). Celui-ci permettra d'augmenter votre niveau de sécurité et de valider que le logiciel est conforme à vos exigences de qualité. Faites-nous ensuite vos retours d'expériences pour contribuer à l'amélioration du logiciel.

Les bogues, cela arrive à tout le monde ! Et si certains d'entre eux sont rapportés, c'est mieux. Alors, en toute logique, ils doivent être triés pour faciliter la tâche des empaqueteurs/développeurs qui doivent valider le bogue (peut-il être reproduit ?), recueillir les infos nécessaires au débogage et assigner le rapport à la bonne personne.
Si vous rencontrez un bogue que vous pouvez reproduire de façon systématique, soumettez un rapport de bogue **en créant un ticket** [34].

Aide aux utilisateurs et support du projet – Vous souhaitez accueillir et aider les nouveaux utilisateurs, ou bien partager des astuces avec les plus expérimentés ? **Intégrez** le Groupe de

Travail RESINFO SWMB [6], des visioconférences sont organisées régulièrement. Contactez le projet via la liste de discussion [35], ou via le chat du projet [36].

Mise à disposition du logiciel – Rendre disponible aisément le logiciel SWMB sur un parc nécessite un petit effort de packaging. Le projet fournit déjà plusieurs exemples de paquets prêts à l'emploi (WAPT, OCS, PDQ, Installateur silencieux). Si votre gestionnaire de paquets préféré est absent, **proposez un paquet SWMB** et sa documentation, pour permettre à d'autres utilisateurs d'utiliser le logiciel facilement, et proposez également le logiciel SWMB pré-packagé sur votre site, en libre usage pour d'autres utilisateurs, ou sur la page de téléchargement du projet [18].

Écriture et documentation – Vous avez le goût de l'écriture agréable, correcte, claire et concise ? Vous aimez relever le défi qui consiste à exprimer clairement des idées ou des systèmes complexes et à enseigner aux autres ? Proposez de la documentation sur les options pas encore couvertes, des tutoriels d'utilisation, des vidéos ou une traduction.

Code – Vous pouvez ajouter vos propres règles à SWMB, mais aussi les partager ensuite (*Tweaks*, Modules et *Presets*), ou améliorer le code du projet. Mettez vos compétences techniques à profit pour le cœur du projet SWMB ! Ajouts, réglages, correctifs et maintenance du logiciel SWMB développé en PowerShell, sont possibles via le Gitlab du projet, en suivant les recommandations du guide du développeur [37].

Demandes de nouvelles fonctionnalités – Pour toute demande de nouvelles fonctionnalités ou de signalement de bug, utilisez le Gitlab Issue du projet [34] ou la liste de diffusion [35].