Sécurisation des postes Windows 10 et 11 via une application modulaire en PowerShell : SWMB

Gabriel Moreau, Olivier De-Marchi

Laboratoire LEGI - UMR5519

24 novembre 2022 / Lille

Origine du projet

- Rejouer l'ANF SIARS `v2.1` en région
- Script PowerShell au LEGI / GPO sur Active Directory à la DR11
- Proposition d'un Groupe de Travail RESINFO réduit à quelques membres en 2019
- Réalisation d'une première maquette de faisabilité en 2020
- Élargissement du groupe de travail
- Prendre contact avec l'ANSSI pour leur signaler le projet
- Laisser l'ASR autonome à 100% sur la politique de sécurité de son unité
- Aider et mutualiser le travail des ASR par une meilleure collaboration

GT SWMB et Comité de pilotage

- Départ de ~~David Gras~~ (DR11 / Grenoble $\Longrightarrow$ CROUS)
- Gabriel Moreau (LEGI / Grenoble)
- **Olivier de Marchi** (LEGI / Grenoble)
- Clément Deiber (DR11 / Grenoble)

- 27 personnes sur la liste SWMB du GT

SWMB (Secure Windows Mode Batch)

- Besoin de sécuriser Microsoft Windows 10 (et 11)
- **Outil modulaire** avec des **règles** et des **anté-règles** (pourvoir faire et défaire)
- Outil sans état « comme » un gestionnaire de configuration sous GNU/Linux (`cfengine, puppet, ansible...`)
- SWMB peut-être lancé plusieurs fois à l'identique
- **Outil en production** au LEGI sur tous les postes
- Pas d'interaction avec l'utilisateur, bien tester sur quelques postes avant de trop déployer
- Packaging pour **simplifier** son propre **déploiement** : `setup.exe` (NSIS), OCS, WAPT, PDQ Deploy

- Ne pas réinventer la roue
- Point de départ, le projet «Win10-Initial-Setup-Script» par Disassembler0.

SWMB - Deux cas particuliers

- Règle particulière pour **supprimer Kaspersky** dans tous les cas (mot de passe Kaspersky et/ou Agent ou pas) $\Longrightarrow$ distribution spécifique de l'outil
- **Chiffrement** des disques (disque système et disque complémentaire avec interaction utilisateur).

SWMB et Active Directory

- SWMB n'est pas incompatible avec l'Active Directory
- Avec SWMB, on sait quelles actions sont lancées et quand
- Il y a presque toujours des machines hors AD dans un parc machine (serveur de badge, automate GTC...). Comment gérez-vous ces machines au cours du temps ?
- SWMB permet de garder dans une **arborescence Git** (GitLab) l'ensemble des configurations au cours du temps et quelle personne a poussé (validé) une modification.
- Fichiers de **configuration au format texte** donc auto-documenté
- Cet ensemble permet de répondre à un objectif de **qualité des règles** sur son parc au **cours du temps**

Vocabulaire - 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.

Organisation du code - Les **modules**

- Le dossier `Modules` regroupe le module principal `SWMB.psm1` qui intègre les routines du cœur des algorithmes
- Ainsi que le module `SWMB.psd1` qui permet de charger tous les modules secondaires
- Les sous modules sont placés dans le sous-dossier `Modules\SWMB`.
- Le code concernant l'implémentation des **tweaks** de l'ordinateur en tant que tel (LocalMachine).
- Le code concernant l'implémentation des **tweaks** de l'utilisateur courant (CurrentUser, extension ≪_CU≫).
- Exemple :
  - `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`

Organisation du code - Les **tweaks**

- Les **tweaks** sont souvent implémentés avec 3 fonctions PowerShell.
- Exemple avec ClearPageFile (nettoyer le fichier `PAGEFILE.SYS` lors de l'arrêt de la machine)
  - `TweakEnableClearPageFile` - tweak **EnableClearPageFile**
  - `TweakDisableClearPageFile` - tweak **DisableClearPageFile**
  - `TweakViewClearPageFile` - voir dans quel état nous sommes (utile pour le débogage)
- Par précaution et sécurité, toutes les fonctions doivent commencer par le préfixe **Tweak**.
- Ainsi, SWMB n'exécute pas n'importe quel code PowerShell.

Exemple des tweaks `ClearPageFile`

```
# ClearPageFileAtShutdown
# https://deployadmin.com/2019/11/03/vider-le-fichier-dechange-a-chaque-arret-de-windows/
# Enable
Function TweakEnableClearPageFile { # RESINFO
  Write-Output "Clear PageFile.sys at shutdown..."
  Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management" -Name "
      ClearPageFileAtShutdown" -Type DWord -Value 1
}

# Disable
Function TweakDisableClearPageFile { # RESINFO
  Write-Output "Do not reset PageFile.sys at shutdown..."
  Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management" -Name "
      ClearPageFileAtShutdown" -Type DWord -Value 0
}

# View
Function TweakViewClearPageFile { # RESINFO
  Write-Output 'Clear PageFile.sys (0 nothing enable, 1 clear at shutdown)'
  $KeyPath = "HKLM:\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management"
  Get-ItemProperty -Path $KeyPath -Name "ClearPageFileAtShutdown"
}
```

Listing 1: clear-page-file.ps1

Type de **tweaks**

- `Enable / Disable`
- `Show / Hide`
- `Install / Uninstall`
- `Add / Remove`
- `Set / Unset`
- `SysMessage, SysRestart, SysRequireAdmin...`
- `View`

- Par défaut, les tweaks n'ont aucun paramètre
- Certains tweaks se configurent via une variable globale
  (Cela permet par exemple de faire passer en paramètre le serveur de temps).
- Les tweaks pour modifier la configuration de l'utilisateur courant finissent par « `_CU` »

Bilan du nombre de **tweaks**

Status	**Number of tweaks**			
Info	Number of RESINFO tweaks			104
Info	Number of Enable and Disable tweaks	182	182	364
Warn	Number of Install and Uninstall tweaks	20	28	48
Warn	Number of Show and Hide tweaks	56	53	109
Info	Number of Add and Remove tweaks	3	3	6
Warn	Number of Set and Unset tweaks	38	7	45
Warn	Number of Pin and Unpin tweaks	0	2	2
Info	Number of total tweaks GPO			574
Info	Number of Sys tweaks (system)			9
Info	Number of View tweaks (debug)			14
Info	Number of Obsolete tweaks			3
Info	Number of total tweaks functions			600

Organisation du code - Les **presets**

- Les **presets** sont des fichiers de configuration à déployer
- Ce sont des listes de tweaks, un par ligne
- # est le caractère de commentaire
- Il y a des exemples de presets qui vérifie le document de l'ANSSI
- Le tweak recommandé est en général écrit en premier sur la ligne

Exemple de fichier de **presets**

```
SysRequireAdmin

### Privacy Tweaks ###
DisableTelemetry                # EnableTelemetry
DisableCortana                  # EnableCortana
DisableActivityHistory          # EnableActivityHistory         # ViewActivityHistory

### UWP Privacy Tweaks ###
DisableUWPBackgroundApps        # EnableUWPBackgroundApps
# DisableUWPVoiceActivation     # EnableUWPVoiceActivation

### Service Tweaks ###
# SetTargetRelease              # UnsetTargetRelease
DisableAppSuggestions           # EnableAppSuggestions          # ViewActivityHistory
# SetP2PUpdateLocal             # SetP2PUpdateInternet          # SetP2PUpdateDisable
DisableDiagTrack                # EnableDiagTrack               # ViewDiagTrack

### UWP Privacy Tweaks ###
DisableUWPBackgroundApps        # EnableUWPBackgroundApps
# DisableUWPNotifications       # EnableUWPNotifications

### Security Tweaks ###
DisableAdminShares              # EnableAdminShares
EnableASLR                      # DisableASLR                   # ViewASLR
# EnableClearPageFile           # DisableClearPageFile          # ViewClearPageFile
```

Classement des **Tweaks** dans les **Modules** et les fichiers de **Presets**

Les tweaks concernant LocalMachine et CurrentUser sont regroupés en quelques grandes catégories (hiérarchie différente de celle des GPO et de l'ANSSI).

- `Application Tweaks`
- `Auxiliary Functions Tweaks`
- `Bitlocker Tweaks`
- `Explorer UI Tweaks`
- `Network Tweaks`
- `Privacy Tweaks`
- `Security Tweaks`
- `Server Specific Tweaks`
- `Service Tweaks`
- `TemporaryBypass Tweaks`
- `UI Tweaks`
- `Unpinning Tweaks`
- `UWP Privacy Tweaks`

Boucle principale - schéma de fonctionnement simplifié

```
# 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
```

Listing 3: main-loop.ps1

Utilisation en ligne de commande

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

- `-import module.psm1` charge le module. Cette option peut-être multiple.
- `-preset file.preset` charge tous les tweaks définis dans le fichier preset. Cette option peut-être multiple.
- `-log msg.log` écrit tous les messages dans le fichier de log.
- `-check` n'exécute pas les tweaks mais vérifie seulement leur existence
- `-exp` est juste un raccourci pour importer le module `Modules/SWMB/Experimental.psm1`.
- `-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.
- ...

Utilisation via les tâches planifiées

Lors de l'installation, SWMB propose de configurer 3 tâches planifiées :

- **LocalMachine-Boot.ps1** - Tâche se lançant au démarrage de la machine. Problème, beaucoup d'utilisateur reboot peu souvent leur machine
- **LocalMachine-PostInstall.ps1** - Tâche se lançant en asynchrone après l'installation. Permet de forcer des réglages de suite
- **CurrentUser-Logon.ps1** - Tâche se lançant à l'ouverture de la session utilisateur

- Avec ce mécanisme de tâches programmées, il y a rarement besoin de lancer SWMB manuellement sur un poste.
- Les paramètres de configuration des tâches programmées sont dans les dossiers `C:\ProgramData\SWMB\*`.

Il y a un mécanisme qui permet de mettre à jour la tâche si vous avez pris la tache par défaut. Sinon, celle-ci ne se met pas à jour lors de l'installation d'une nouvelle version.

Tâche programmée se lançant lors du boot

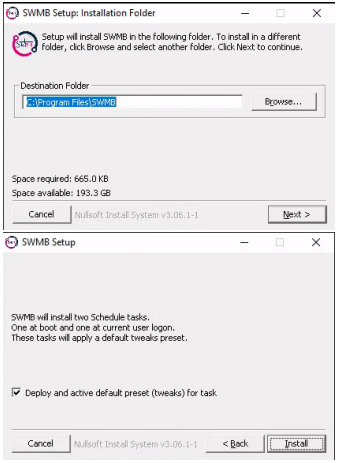
```
# Define Boot preset on ProgramData
$DataFolder  = "$Env:ProgramData/SWMB"
$BootPreset  = "$DataFolder/Presets/LocalMachine-Boot.preset"
$BootModule  = "$DataFolder/Modules/LocalMachine-Boot.psm1"
$BootLog     = "$DataFolder/Logs/LocalMachine-LastBoot.log"
$BootHash    = "$DataFolder/Caches/LocalMachine-LastBoot.hash"

# Revert if not exist to default module name Local-Addon.psm1
If (!(Test-Path -LiteralPath $BootModule)) {
  $BootModule = "$DataFolder/Modules/Local-Addon.psm1"
}
# Launch SWMB with this preset
If (Test-Path -LiteralPath $BootPreset) {
  If (Test-Path -LiteralPath $BootModule) {
    .\swmb.ps1 -log "$BootLog" -import "$BootModule" -preset "$BootPreset" -hash $BootHash
  } Else {
    .\swmb.ps1 -log "$BootLog" -preset "$BootPreset" -hash $BootHash
  }
}
```

Listing 4: LocalMachine-Boot.ps1

Installation graphique ou silencieuse

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

- Installeur NSIS (fabriqué par intégration continue GitLab sous Debian)
- « 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`)



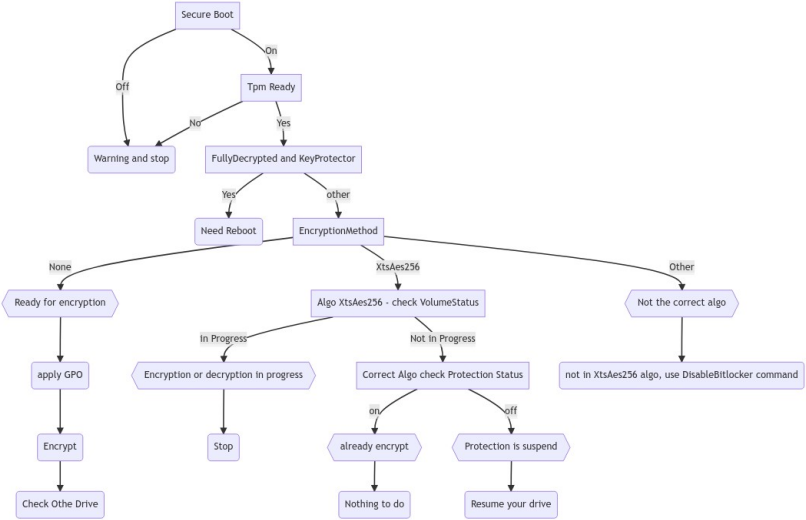
Interface graphique minimaliste

- Lancer le script interactif de chiffrement des lecteurs avec BitLocker
- 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.

Chiffrement BitLocker

- `.\swmb.ps1 EnableBitlocker # DisableBitlocker`
- Ordinateur configuré en mode de démarrage UEFI avec Secure Boot
- Puce TPM dans l'état « prête »
- Utilisation de l'algorithme de chiffrement XtsAes256
- Proposition de chiffrement de tous les disques internes
- Déchiffrement automatique des disques non système (D, E...)
- Sauvegarde des clefs de chiffrement, en local et/ou sur un lecteur réseau
- Possibilité de mettre un code PIN (préférable)
- Prise en compte de l'état initial du poste (ordinateur déjà chiffré, chiffrement en cours...).

Chiffrement BitLocker - Algorithme



Chiffrement BitLocker

- Avant le chiffrement, BitLocker est configuré via `HKLM:\SOFTWARE\Policies\Microsoft\FVE`
  - Forcer l'algorithme de chiffrement XtsAes256, `EncryptionMethodWithXtsOs = 7`
  - Interdire à l'utilisateur de modifier le code PIN, `DisallowStandardUserPINReset = 1`
  - ...
- Clefs de chiffrement / déchiffrement stockés sur le disque système
- Des droits particuliers sont appliqués sur les fichiers contenant ces clefs (lecture impossible, copie par un compte administrateur)
- À charge à chacun de sauver ces clefs dans un coffre-fort centralisé (container VeraCryt par exemple)

SWLN : Secure Windows Local Network

- Comment utiliser SWMB dans son unité, sur mon site ?
- Déployer SWMB tel quel en appliquant le jeu de preset par défaut du GT RESINFO (règles de l'ANSSI plus quelques autres)
- Étendre SWMB
  - En effet, SWMB est un framework qui exécute des fonctions PowerShell en CLI ou dans des tâches programmées
  - Écrire son code spécifique sous forme de fonctions TweakEnable, TweakSet... dans un module
  - L'écriture des anté-fonctions n'est pas obligatoire chez soi ! (préférable sur le projet SWMB)

SWLN : Secure Windows Local Network

- Placer votre code dans le module `C:\ProgramData\Modules\Local-Addon.psm1`
- Mettre du code spécifique (non PowerShell par exemple) dans `C:\ProgramFile\SWLN\`
- Écrire ses jeux de presets (la commande spéciale `$PRESET` permet d'importer un autre jeu de preset, dont par exemple le jeu par défaut.
- Intégrer l'installeur SWMB dans votre code
- Faire un installeur `setup.exe` de votre extension SWLN
- Déployer SWLN (SWLN doit être installé avant SWMB - configuration des tâches programmées, notamment la `post-install`)
- Remonter dans SWMB votre code qui vous semble mutualisable

SWLN au LEGI : LabStumm

- `CurrentUser-Logon.preset`
- `LocalMachine-Boot.preset`
- `LocalMachine-PostInstall.preset`
- `Local-Addon.psm1`
- `Custom-VarOverload.psm1` - paramètre global pour quelques tweaks
- `install.bat`
- `post-install.ps1`
- `uninstall.bat`
- `Makefile` - créer le Zip qui va bien pour OCS
- `print` - dossier avec les drivers des photocopieurs
- `SpeculationControl` - dossiers avec quelques outils pour tester si les mitigations de type spectre sont actives
- `fr-oss` - dossier permettant d'avoir un clavier fr-oss (comme sous GNU/Linux)

Exemple de paramètres globaux

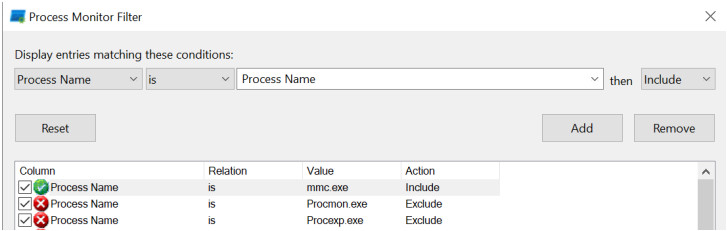
```
# NTP
$Global:SWMB_Custom.NTP_ManualPeerList = 'XXX.XXX.XXX.XXX'

# Target Release
$Global:SWMB_Custom.ProductVersion           = 'Windows 10'
$Global:SWMB_Custom.TargetReleaseVersionInfo = '21H2'
```

Listing 5: Custom-VarOverload.psm1

Ajouter ses propres règles

- En général, un tweak (ou une GPO) revient à modifier la valeur d'une clef de registre
- 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 (`gpsearch`)
  - Group Policy Administrative Templates Catalog (`admx.help`)
  - Tableur Excel de Microsoft contenant toutes les GPO des systèmes d'exploitation
- On trouve presque toujours la solution sur internet et sinon...

Ajouter ses propres règles

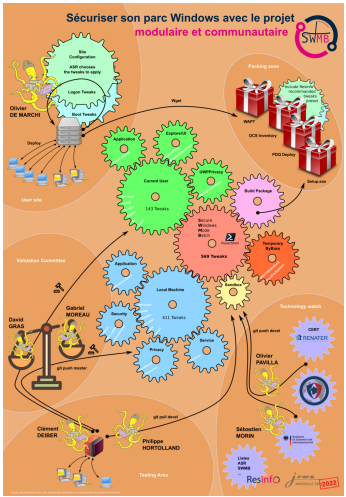


Capture de la clef de registre avec ProcessMonitor de Sysinternals

Aventure JRES Marseille 2022

Titre : **Sécuriser son parc Windows avec le projet modulaire et communautaire SWMB**

- 6 mois de travail collectif
- Visio-conférence tous les 15 jours
- Un poster
- Un article de 30 pages
- Olivier De-Marchi, Clément Deiber, Gabriel Moreau, David Gras, Philippe Hortolland, Olivier Pavilla, Sébastien Morin



Conclusion - **SWMB**

- **Programme libre**, **modulaire** et **collaboratif**
- Chaque ASR l'adapte à son contexte et ne pousse que les tweaks qu'il souhaite (**autonomie**, rien n'est obligatoire)
- **Fonctionne** en production
- Il est facile de modifier ses propres scripts PowerShell pour les intégrer dans cet environnement
- Le projet à besoin des ASR pour progresser et intégrer des nouvelles fonctionalités
- Prenez **votre sécurité en main** en **partageant** aussi votre **savoir faire**

- SWMB n'a pas les mêmes objectifs que tous les programmes graphiques qui vous proposent des tweaks

Licence

Merci à toutes les personnes et entités
nous ayant aidés ou ayant participé depuis le début

Cette présentation est sous : Licence Art Libre

http://artlibre.org/