

1. Projet d'entreprise

Présentation du projet

Le projet consiste en la création et l'optimisation d'un outil visant à faciliter la récupération des données utilisateur sur un poste informatique sous Microsoft Windows. Ce processus est essentiel lors du renouvellement ou de la réinstallation d'un poste, où les données doivent être transférées efficacement et en toute sécurité.

Ce projet aurait parfaitement pu être réalisé en méthode Agile, il a été décidé d'opter pour la méthode en Cascade. En effet, le projet est destiné à une utilisation interne au sein du service SGP. En l'absence de client externe, il a été développé pour répondre aux besoins opérationnels spécifiques de notre équipe. En contrepartie, le fait de suivre un processus Agile avec des sprints aurait impliqué des cycles de réévaluation et d'ajustement, ce qui aurait entraîné une perte de temps en l'absence de client. Comme le cahier des charges était clairement défini dès le départ, la méthode Cascade, avec son approche séquentielle et linéaire, s'est révélée plus appropriée sans qu'il y ait de retour en arrière fréquents suivant une méthode Agile.

Le projet a donc été suivi selon la méthode Cascade qui vise à séquencer les différentes étapes du projet à l'avance afin de les suivre une à une. Une représentation de ces étapes est disponible à la suite de la présentation (page 16).

Contexte du projet

Lors du remplacement ou de la réinstallation d'un poste informatique, il est crucial de garantir que les données des utilisateurs soient correctement transférées. Les sauvegardes traditionnelles peuvent ne pas refléter les données les plus récentes (potentiellement des données jusqu'à J-1, voir plus en cas de période de fermeture). Ce projet vise à pallier ce problème en mettant en place une solution capable de récupérer les données directement à partir du poste en cours, assurant ainsi que les informations soient à jour au moment du transfert.

Objectifs et Problématique du projet

L'objectif principal de ce projet est de simplifier et d'automatiser le processus de récupération des données utilisateurs sur des postes Windows, notamment lors des réinstallations ou renouvellements de machines. Actuellement, ces données sont transférées manuellement ou à partir de sauvegardes qui peuvent ne pas être à jour, ce qui entraîne des erreurs et des pertes de temps.

En outre, la problématique réside dans la diversité des configurations des postes Windows au sein de l'organisation, notamment en termes de matériel et de protection des données (chiffrement BitLocker). Le projet doit donc proposer une solution capable de s'adapter à ces variables tout en assurant un transfert de données automatisé, fiable et sécurisé.

Ensuite, la solution doit être capable de fonctionner sur tout type de poste Windows, qu'il s'agisse de machines récentes ou plus anciennes, avec des configurations variées. Elle doit également pouvoir être exécutée à partir d'un démarrage réseau, garantir la sécurité des données pendant leur transfert, et fonctionner sans intervention humaine prolongée une fois lancée. De plus, le déchiffrement des disques protégés par BitLocker doit être pris en compte, afin de rendre les données accessibles pour leur sauvegarde. Finalement, le projet vise à optimiser le processus en réduisant le temps consacré à la récupération des données.

Cahier des Charges :

- **Démarrage Réseau** : L'outil doit être opérationnel dès le démarrage du poste via un boot réseau, permettant une récupération efficace sans dépendre de systèmes préexistants.
- **Déchiffrement du Disque** : L'outil doit être capable de déchiffrer les disques protégés par BitLocker pour accéder aux données, garantissant ainsi que les informations chiffrées soient récupérées sans compromettre leur sécurité.
- **Sauvegarde des Données** : L'outil doit permettre de sauvegarder les données utilisateurs au moment précis du transfert, assurant la capture des informations les plus actuelles.

- **Stockage Temporaire** : Les données doivent être stockées temporairement sur un serveur avant d'être réinjectées dans le nouveau poste ou le poste réinstallé. Cette étape est essentielle pour la gestion des données durant le processus de transfert.

Démarche : Pour atteindre ces objectifs, une analyse approfondie des outils existants et des technologies disponibles est nécessaire. L'outil pourrait soit utiliser des solutions déjà établies adaptées aux besoins, soit être développé sur mesure en combinant des technologies comme PXE (Preboot Execution Environment) avec l'outil iPXE, WinPE (Windows Preinstallation Environment) et PowerShell, permettant une personnalisation complète du processus de récupération.

Solution Proposée

Il existe plusieurs solutions pour la récupération de profils utilisateurs, comme Microsoft User State Migration Tool (USMT), Clonezilla, ou Dell Data Protection | Endpoint Recovery. Cependant, ces outils peuvent être payants, ou comme Dell Endpoint Recovery, limités à des machines de la marque Dell. En revanche, un script personnalisé offre une solution plus flexible et économique, adaptable à tout type de poste, sans les contraintes des solutions propriétaires.

Justification de l'Utilisation d'un Script :

1. **Flexibilité** (Adapté selon le besoin)
2. **Gain de Temps** (Réduction du temps d'intervention)
3. **Économique** (Pas de coût de licence)
4. **Simplicité d'Utilisation** (Facile à exécuter et personnaliser)

En résumé, l'adoption de ce script automatisé est justifiée par sa capacité à améliorer l'efficacité du processus de récupération des données, à réduire le risque d'erreurs humaines, et à accélérer le temps de traitement, tout en assurant la sécurité des données et la simplicité d'utilisation.

La solution développée pour le projet utilise la méthode Cascade, ce qui a permis de définir les étapes du projet de manière séquentielle dès le départ. Cette approche a facilité une planification claire et structurée, avec des étapes préalablement établies pour assurer une réalisation efficace du projet. Les étapes définies pour le projet sont les suivantes :

Étape	Test	Résultat
1. Création d'un environnement WinPE	En attente	En attente
1.1 Ajout des modules PowerShell, ActiveDirectory et BitLocker	En attente	En attente
2. Rédaction du fichier de configuration startnet	En attente	En attente
3. Rédaction du script recuperation_profil	En attente	En attente
3.1 - Définition des identifiants de connexion	En attente	En attente
3.2 - Récupération du numéro de l'ordinateur	En attente	En attente
3.3 - Récupération de la clé BitLocker dans l'AD puis déchiffrement du disque	En attente	En attente
3.4 - Connexion au serveur Servinst	En attente	En attente
3.5 - Liste des profils à récupérer avec une liste box	En attente	En attente
3.6 - Copie des profils listés sur servinst	En attente	En attente
3.7 - Création d'une clé USB bootable pour les tests	En attente	En attente
4. Montage du projet sur le serveur	En attente	En attente

Présentation de la solution

Avant de pouvoir exécuter le script de récupération des profils, il est essentiel de configurer un environnement WinPE adapté.

La rédaction du script ne sera pas détaillée en détail dans le rapport. Les étapes du projet abordées seront la création de l'image WinPE, ainsi que la configuration de son environnement, la rédaction d'un fichier de configuration ainsi que l'utilisation du projet. Des documentations sources seront ajoutées tout du long de l'explication du projet, celles-ci comportent les étapes et les lignes de commande précises lors de l'installation et la configuration de WinPE pour une compréhension plus approfondie.

Cependant, avant d'exécuter le script de récupération, il est crucial de créer et de configurer l'environnement WinPE. En effet, la première grande étape de la solution apportée inclut la création de l'image WinPE, ainsi que l'ajout des modules nécessaires (PowerShell, Active Directory, BitLocker), la configuration de la langue française et la création du fichier de configuration startnet.cmd.

Configuration de l'Environnement WinPE

1. Création d'un Environnement WinPE

L'objectif consiste à créer un environnement WinPE, essentiel pour exécuter le script de récupération des profils utilisateurs. WinPE est une version allégée de Windows utilisée principalement pour les installations, les réparations et les déploiements. (Voir la documentation dans le Glossaire)

Téléchargement et Installation de l'ADK et WinPE

Pour commencer, il est nécessaire de télécharger et d'installer Windows Automated Installation Kit (ADK) et son composant additionnel Windows PE. Ces outils permettent la création et la gestion d'images Windows. Les liens de téléchargements se trouvent dans le glossaire.

- Windows ADK (version 10.1.26100.1, mai 2024)
- Composant additionnel Windows PE pour Windows ADK (version 10.1.26100.1, mai 2024)

Une fois ces packages installés, nous disposons d'un outil de commande appelé Environnement de déploiement et d'outils de création d'images, qui est crucial pour la création de l'image WinPE. (Voir son utilisation ainsi que les commandes utilisées pour sa configuration en annexe page 39)



Création de l'image WinPE

Une fois les outils installés, il est possible de procéder à la création de l'image WinPE à l'aide du logiciel installé ci-dessus. Cette image permet d'exécuter un environnement léger de Windows directement à partir d'un périphérique USB ou d'un réseau.

Une documentation Microsoft de l'installation exacte de WinPE est située dans le glossaire.

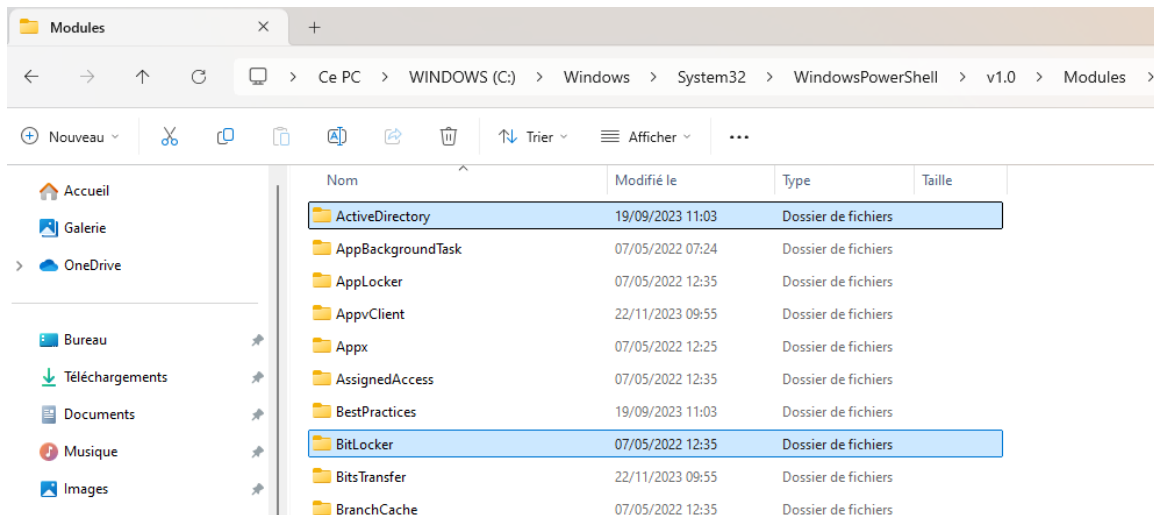
1.1 Configuration de WinPE

Une fois l'environnement WinPE créé, il est nécessaire d'ajouter certains modules spécifiques pour l'utilisation du script :

- **Module PowerShell** : Ce module permet l'automatisation des tâches dans WinPE, tel que l'exécution du script et l'intégration des modules suivants.
- **Modules Active Directory et BitLocker** : Ces deux composants sont essentiels pour se connecter à l'Active Directory afin de récupérer la clé BitLocker, sans ces composants, il serait impossible d'authentifier les utilisateurs ou de déchiffrer les disques, bloquant ainsi l'accès aux données critiques que le script est censé récupérer.

Les modules Active Directory et BitLocker sont des composants de PowerShell qui se trouvent au chemin d'accès suivant :

C:\Windows\System32\WindowsPowerShell\v1.0\Modules



L'installation de ces deux modules seront exceptionnellement détaillées suite à l'absence de documentation sur internet :

```

Fichier Edition Format Affichage Aide
### Modules ActiveDirectory et BitLocker :
xcopy /E /I "C:\Windows\System32\WindowsPowerShell\v1.0\Modules\ActiveDirectory"
"C:\WinPE_amd64_BL\mount\Windows\System32\WindowsPowerShell\v1.0\Modules\ActiveDirectory"

xcopy /E /I "C:\Windows\System32\WindowsPowerShell\v1.0\Modules\BitLocker"
"C:\WinPE_amd64_BL\mount\Windows\System32\WindowsPowerShell\v1.0\Modules\BitLocker"

xcopy "C:\Windows\Microsoft.NET\assembly\GAC_64\Microsoft.ActiveDirectory.Management.resources\v4.0.10.0.0_fr_31bf3856ad364e35\Microsoft.ActiveDirectory.Management.resources.dll" "C:\WinPE_amd64_BL\mount\Windows\System32"

xcopy "C:\Windows\Microsoft.NET\assembly\GAC_64\Microsoft.ActiveDirectory.Management\v4.0.10.0.0_31bf3856ad364e35\Microsoft.ActiveDirectory.Management.dll" "C:\WinPE_amd64_BL\mount\Windows\System32\WindowsPowerShell\v1.0\Modules\ActiveDirectory"

dism /image:C:\winpe_amd64_BL\mount /add-package /packagepath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\WinPE-WMI.cab"

dism /image:C:\winpe_amd64_BL\mount /add-package /packagepath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\WinPE-SecureStartup.cab"

dism /image:C:\winpe_amd64_BL\mount /add-package /packagepath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\WinPE-EnhancedStorage.cab"
  
```

Tout d'abord, les commandes xcopy servent à copier les modules ActiveDirectory et BitLocker, depuis le système vers l'image WinPE montée. Par exemple, la première commande copie de manière récursive le répertoire ActiveDirectory vers l'image WinPE. Ensuite, des fichiers DLL (Dynamic Link Library, bibliothèque de fonctions/ressources) sont copiés dans l'image WinPE, car ils sont nécessaires pour l'utilisation des commandes Active Directory.

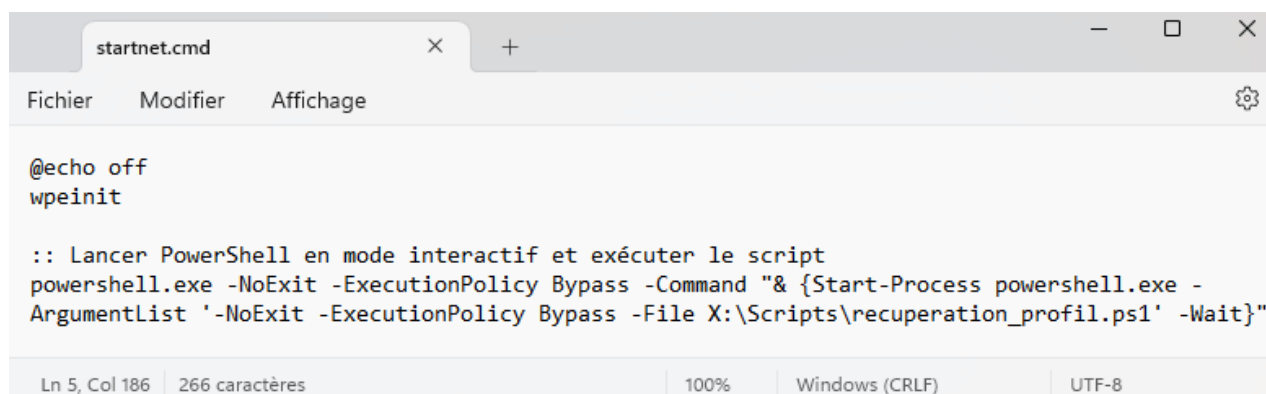
Ensuite, les commandes DISM (Deployment Image Servicing and Management) permettent d'ajouter des packages essentiels comme WinPE-WMI.cab active le support de WMI (Windows Management Instrumentation), WinPE-SecureStartup (gestion BitLocker), et WinPE-EnhancedStorage (gestion des disques chiffrés), assurant que l'image est prête pour la suite des opérations.

- **Module de la langue française et de compatibilité matérielle :**

Après l'ajoute des modules principaux, il est essentiel d'adapter l'environnement à notre configuration spécifique. Cela se fait principalement en intégrant le pack de langue française, afin de garantir une interface adaptée, ainsi que des pilotes matériels pour assurer la compatibilité avec une large gamme de dispositifs.

2. Rédaction du fichier de configuration startnet.cmd

Afin d'automatiser le démarrage du script au démarrage de WinPE, il est nécessaire de créer le fichier nommé **startnet.cmd**. Ce fichier est essentiel car il permet d'exécuter notre script principal avec PowerShell ainsi que de configurer WinPE.



```
@echo off
wpeinit

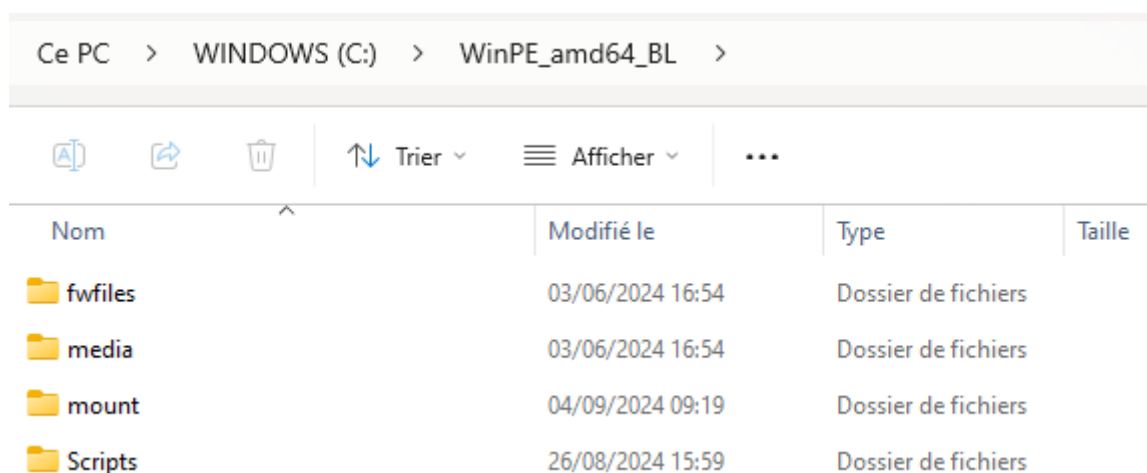
:: Lancer PowerShell en mode interactif et exécuter le script
powershell.exe -NoExit -ExecutionPolicy Bypass -Command "& {Start-Process powershell.exe -
ArgumentList '-NoExit -ExecutionPolicy Bypass -File X:\Scripts\recuperation_profil.ps1' -Wait}"
```

Dans ce fichier, la commande « wpeinit » permet d'initialiser les services WinPE. Ensuite, la commande powershell.exe ouvre PowerShell suivi du script recuperation_profil.ps1 avec les bons paramètres. Nous utilisons l'option « -ExecutionPolicy Bypass » pour permettre au script de s'exécuter sans restriction. De la même manière, les liens sources qui détaillent ce fichier sont disponibles dans le glossaire.

3. Rédaction du script recuperation_profil

Avant de commencer, le dossier où nous allons monter l'image WinPE, C:\WinPE_amd64_BL\mount, est vide. En montant l'image WinPE dans ce dossier, nous accédons directement à son système de fichiers, ce qui nous permet d'ajouter des fichiers nécessaires, comme notre script de récupération de profils ainsi que les modules importés.

Initialement, le script se trouve dans le répertoire C:\WinPE_amd64_BL\Scripts. Suite au montage de l'image WinPE, ce script est placé dans C:\WinPE_amd64_BL\mount\Scripts. Lorsque WinPE est en cours d'exécution, ce répertoire apparaît sous la lettre de lecteur X:, ce qui signifie que le chemin X:\Scripts\recuperation_profil fait référence au dossier de montage C:\WinPE_amd64_BL\mount\Scripts\recuperation_profil.



Ce PC > WINDOWS (C:) > WinPE_amd64_BL >			
🔍 🔗 🗑️ ⬆️ Trier ▾ ≡ Afficher ▾ ⋮			
Nom	Modifié le	Type	Taille
📁 fwfiles	03/06/2024 16:54	Dossier de fichiers	
📁 media	03/06/2024 16:54	Dossier de fichiers	
📁 mount	04/09/2024 09:19	Dossier de fichiers	
📁 Scripts	26/08/2024 15:59	Dossier de fichiers	

Une fois que la configuration de WinPE est terminée, nous devons également évaluer les ressources temporelles et budgétaires investies dans ce projet avant de présenter son utilisation.

4. Montage du projet sur le serveur

Après avoir validé le script de récupération des profils sur une clé USB, il est crucial de déployer le projet sur le serveur pour une utilisation pratique. En effet, ce déploiement permet d'éviter la dépendance aux clés USB et facilite l'accès au script via le réseau.

Effectivement, la configuration sur le serveur inclut le transfert des fichiers nécessaires de WinPE et la configuration des options de démarrage réseau avec iPXE. Une fois le projet monté et configuré, nous détaillerons l'utilisation du projet, incluant le démarrage réseau et la sélection du script depuis l'interface.

Transfert des Fichiers sur le Serveur :

- Les fichiers essentiels pour le démarrage de WinPE (wimboot, BCD, boot.sdi, et boot.wim) doivent être transférés sur le serveur dans le répertoire approprié, généralement /var/www/html. Ce répertoire permet aux clients d'accéder aux fichiers via HTTP.
 - wimboot : Chargeur qui démarre l'image WinPE.
 - BCD : Fichier de configuration de démarrage.
 - boot.sdi : Disque virtuel préparant le système pour le démarrage.
 - boot.wim : Image contenant l'environnement WinPE.

Mise en Place du Menu de Démarrage :

- Il est important de configurer le fichier de démarrage réseau pour pointer vers les fichiers nécessaires sur le serveur. Ce fichier de configuration, qui est démarré via l'iPXE, inclut des lignes indiquant l'emplacement des fichiers vu précédemment sur le serveur.

Ressources temporelles et budgétaires du projet

Étant donné que ce projet est une initiative interne sans client externe ni coûts matériels, il est difficile d'en définir un budget précis. De plus, sachant qu'il n'y a eu aucun achat d'équipement ou coûts externes à considérer. Par conséquent, nous nous concentrons sur le temps de travail consacré et le temps gagné grâce à l'automatisation pour évaluer la rentabilité. En effet, en se basant sur le SMIC horaire, nous avons calculé le coût des heures travaillées et les économies réalisées, ce qui permet de mesurer l'impact du projet en termes de productivité.

Le projet s'étend sur une période de 15 semaines en entreprise sur 5 mois. Pendant ces 15 semaines, prenons une moyenne de 12h de travail par semaine consacrées au projet, ce qui totalise 156 heures de travail.

Dans cet exemple, prenons un smic comme point de repère pour les calculs.

Calcul du coût du projet :

Temps total impliqué : 156h

Taux horaire brut SMIC : 1 747 € / 151,67 heures = environ 11,52 € par heure

Coût total du projet : $156 * 11,52€ = 1\,797,12€$

Il est important de noter qu'aucun coût matériel n'a été impliqué dans ce projet. Ainsi, le coût du projet total en se basant sur un SMIC est de 1 797€.

Utilisation du projet

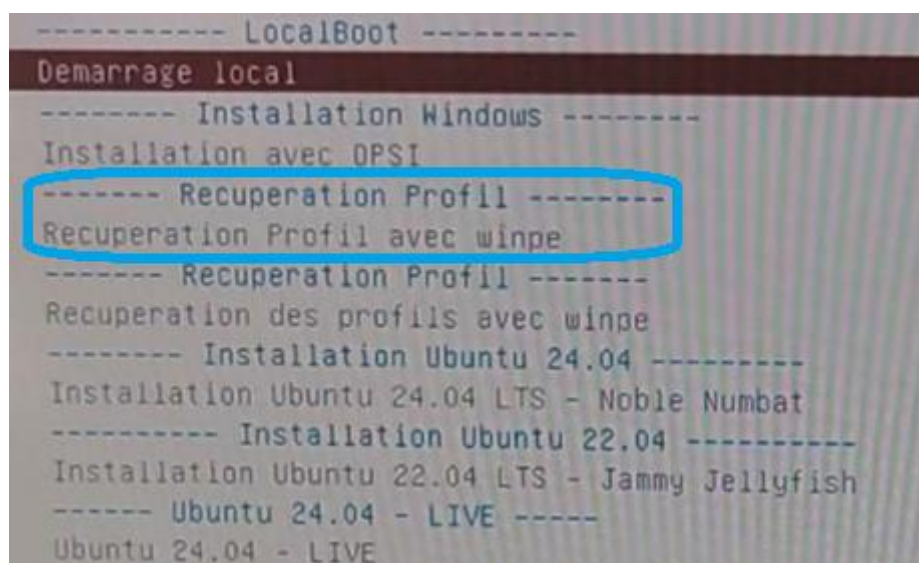
Ensuite, cette partie décrit l'utilisation du script. De plus, la procédure d'exécution du script est disponible en annexe (page 38). Afin d'exécuter le script, nous passons par une technologie de démarrage réseau supportant des menus de démarrage personnalisés ainsi que les scripts, iPXE, se basant sur PXE (Preboot Execution Environment).

En effet, le boot PXE est un protocole de démarrage réseau qui permet à un ordinateur local de démarrer à partir de données situées sur un serveur distant, plutôt que de démarrer à partir du disque local, ou du contenu de la clé USB. Cette méthode de démarrage est souvent utilisée pour le déploiement de postes de travail puisqu'elle va permettre de distribuer le système d'exploitation à installer à partir du réseau.

De plus, le boot iPXE (intelligent Preboot Execution Environment) est un logiciel qui se base sur la technologie PXE. Il permet à un ordinateur de démarrer sur le réseau et ainsi offrir des fonctionnalités supplémentaires telles que :

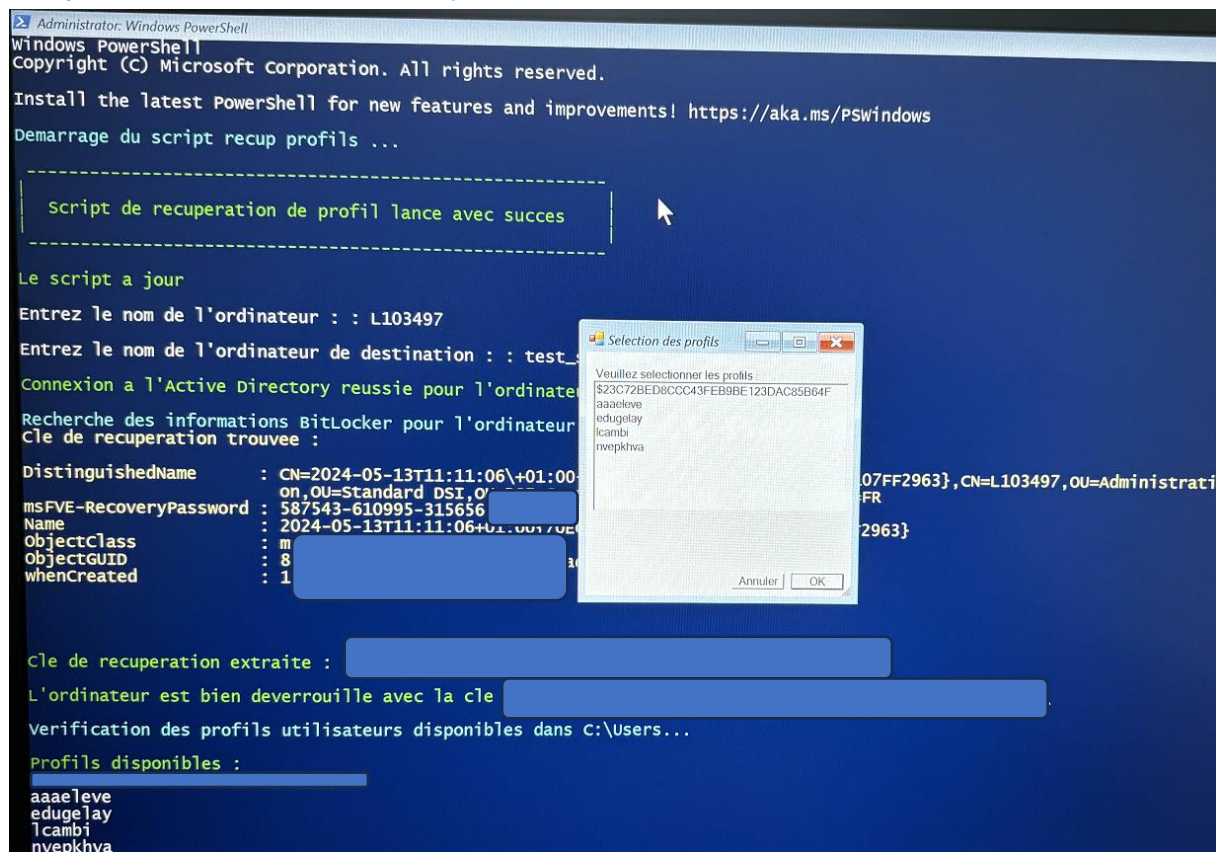
- Le support des scripts
- Le support des menus de démarrage personnalisés

Ensuite, suite au démarrage réseau iPXE, une interface se présente avec différentes possibilités, l'installation de poste avec OPSI (Open PC Server Integration), l'installation d'Ubuntu, ainsi que le script pour récupérer les profils.



Par la suite, le script est exécuté, il est ainsi soumis à des tests de vérifications afin de garantir son fonctionnement. Le script est conçu pour apporter un message d'avertissement à chaque étape lors de la connexion, ces messages permettent au technicien utilisant le script d'avoir un suivi total en cas d'erreur.

Interface lors de l'exécution du script :



Le déroulement du script est le suivant :

- Tout d'abord une interface PowerShell se lance comme vu précédemment, il est ensuite affiché un message de départ
- Ensuite, le script demande le nom de l'ordinateur host et de destination. En effet, l'environnement WinPE n'a pas accès à Windows donc il ne peut pas récupérer seul le nom de l'host. Le nommage des postes se font de la manière suivante : L10XXXX, ici L103497, un portable HP EliteBook 840, et prenons « test_script_winpe » comme nom de destination.

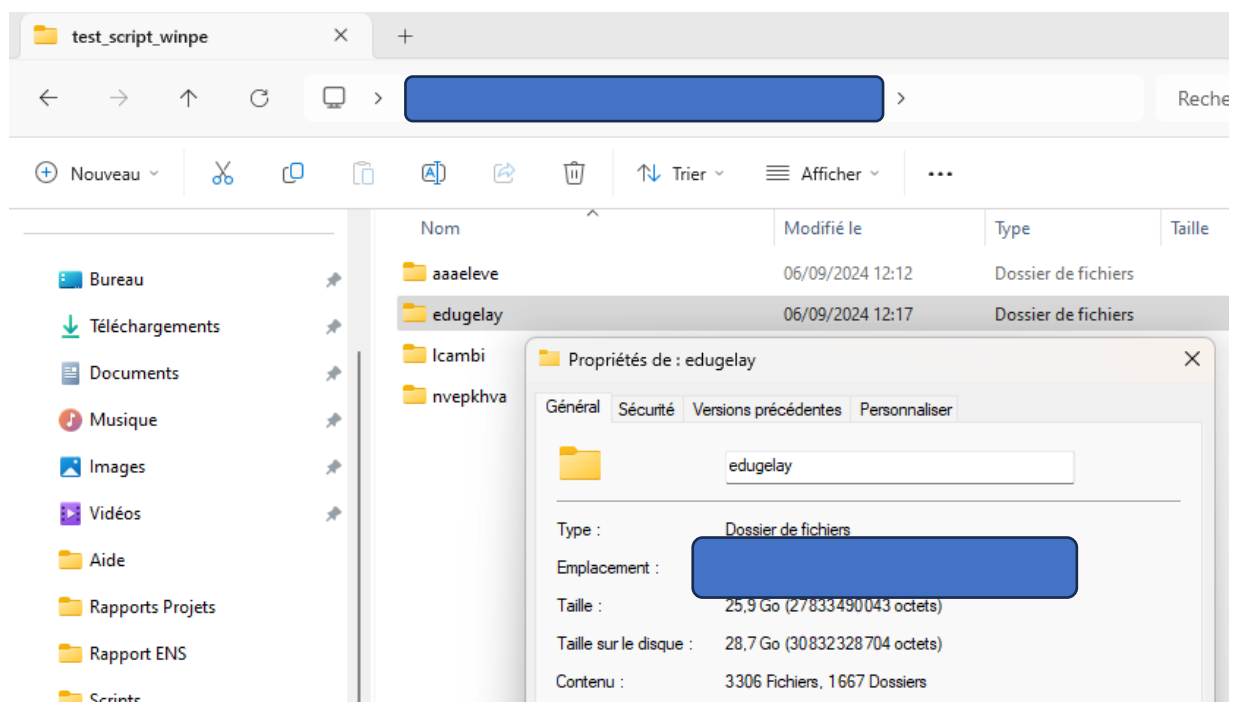
- De plus, le script se connecte à l'Active Directory et recherche ainsi la clé BitLocker liée à l'ordinateur demandé
- Suite à l'obtention de la clé, le script déverrouille l'ordinateur afin d'accéder aux profils.

Certains profils ont été enlevés lors de la récupération, par exemple tous les profils administrateurs, ainsi que les profils « Public », « Default », « All Users » et « Default User ».

- Une listbox est ensuite affichée afin de pouvoir sélectionner les profils voulus.
- Par la suite, un message indiquant que la connexion au serveur est réussie, ainsi que la copie des profils sélectionnés lors de la listbox démarre. Ensuite, la création du dossier « edugelay », correspondant à mon profil, est réussie.

```
Connexion a \ [redacted] reussie
Creation du dossier test sur Servinst reussie.
Source : C:\Users\edugelay
Destination : M:\test\edugelay
Creation du dossier M:\test\edugelay reussie.
```

Suite à l'exécution du script ainsi que la copie des profils concernés, il est possible de vérifier leur copie sur le serveur :



Ainsi, les profils utilisateurs sont copiés de manière efficace et fiable grâce à l'automatisation du script. Le processus est rapide, et ne nécessite qu'une intervention manuelle pour le nom d'ordinateur host et de destination, ce qui réduit les erreurs humaines et assure que l'intégralité des données nécessaires est correctement transférée. De plus, la vérification des logs permet de confirmer la réussite de chaque opération de copie.

En effet, il est possible de vérifier le fonctionnement du projet grâce aux étapes prédéfinies avec l'utilisation de la méthode Cascade :

Étape	Test	Résultat
1. Création d'un environnement WinPE	OK	OK
1.1 Ajout des modules PowerShell, ActiveDirectory et BitLocker	OK	OK
2. Rédaction du fichier de configuration startnet	OK	OK
3. Rédaction du script recuperation_profil	OK	OK
3.1 - Définition des identifiants de connexion	OK	OK
3.2 - Récupération du numéro de l'ordinateur	OK	OK
3.3 - Récupération de la clé BitLocker dans l'AD puis déchiffrement du disque	OK	OK
3.4 - Connexion au serveur Servinst	OK	OK
3.5 - Liste des profils à récupérer avec une liste box	OK	OK
3.6 - Copie des profils listés sur servinst	OK	OK
3.7 - Création d'une clé USB bootable pour les tests	OK	OK
4. Montage du projet sur le serveur	OK	OK

Bénéfices du projet

Le script est utilisé en moyenne 8 fois par semaine. Grâce à son efficacité, il permet de gagner environ 2 heures et 40 minutes par semaine (soit 20 minutes par utilisation). En moyenne, ce gain de temps représente environ 10 heures et 40 minutes par mois.

Calcul du Temps et Économies

De la même manière que pour les ressources budgétaires du projet, sur la base du SMIC brut en France (1 747 € par mois en 2024 pour une durée complète de travail), les économies générées par le script peuvent être calculées comme suit :

Temps gagné par semaine : 2h40

Temps gagné par mois : 10h40 (2h40 * 4)

Taux horaire brut SMIC : 1 747 € / 151,67 heures = environ 11,52 € par heure

Économies mensuelles : 10h40 * 11,52 € = 122,21 € (arrondi)

Le script permet donc de réaliser une économie d'environ 122,21 € par mois en termes de coût horaire basé sur le SMIC.

En conclusion, avec un coût total du projet de 1 797€, le projet sera rentabilisé après environ 15 mois à utilisation moyenne, soit 8 fois par semaine.

Par ailleurs, il est envisageable que pendant certaines périodes de l'année, le script peut être utilisé jusqu'à 15 fois par semaine, ce qui générerait une économie de 225 € par mois. Dans ce scénario, le projet serait amorti au bout d'un an, en supposant une utilisation maximale pendant 3 mois.

Améliorations futures du projet

La suite logique du projet serait ensuite d'automatiser la copie des profils vers l'ordinateur de destination. En effet, le script copie sur le serveur les profils dans le dossier `\nom_d'ordinateur_de_destination\`, cependant le projet n'inclut pas la copie de ces profils vers cet ordinateur.

Une autre idée serait d'ajouter une fonctionnalité au script pour automatiser le chiffrement des disques avec BitLocker. Ainsi, les ordinateurs non chiffrés seraient automatiquement sécurisés, la remontée dans l'Active Directory des clés de récupération BitLocker pourrait aussi être automatisée. Après le chiffrement, le script s'exécuterait pour récupérer et transférer les profils.

Conclusion du projet

Pour conclure, le projet a atteint ses objectifs avec succès, avec toutes les fonctionnalités entièrement opérationnelles. Il m'a offert l'occasion d'explorer et de maîtriser diverses technologies telles que iPXE et PXE, WinPE, ainsi que les scripts PowerShell pour l'automatisation. Cette expérience a enrichi mes compétences techniques en scripting, ainsi qu'en gestion des données et des systèmes, tout en approfondissant ma compréhension des environnements de déploiement et de récupération.

2. Glossaire technique et liens sources

Description des termes techniques utilisés :

- ENS : École Normale Supérieure
- RPI : Responsable de Projets Informatiques
- SGP : Support et Gestion de Parc
- OPSI : Open PC Server Integration
- TOIP :
- PXE : Preboot Execution Environment
- iPXE : intelligent Preboot Execution Environment
- WinPE : Windows Preinstallation Environment
- GLPI : Gestionnaire Libre de Parc Informatique
- ADK : Automated Installation Kit
- DLL : Dynamic Link Library
- DISM : Deployment Image Servicing and Management
- WMI : Windows Management Instrumentation

Liens sources utilisés :

Installer l'ADK et WinPE :

Documentation WinPE : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/winpe-intro?view=windows-11>

Téléchargement de l'ADK : <https://learn.microsoft.com/fr-fr/windows-hardware/get-started/adk-install>

Téléchargement de WinPE : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/download-winpe--windows-pe?view=windows-11#get-the-files-you-need-to-create-winpe-media>

Création et configuration de WinPE :

Création d'un média WinPE : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/winpe-create-usb-bootable-drive?view=windows-11>

Ajout du module PowerShell : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/winpe-adding-powershell-support-to-windows-pe?view=windows-11>

Ajout des drivers à WinPE : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/add-and-remove-drivers-to-an-offline-windows-image?view=windows-11>

Monter et personnaliser WinPE : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/winpe-mount-and-customize?view=windows-11#add-updates-to-winpe-if-needed>

Script de démarrage WinPE avec startnet.cmd : <https://learn.microsoft.com/fr-fr/windows-hardware/manufacture/desktop/wpeinit-and-startnetcmd-using-winpe-startup-scripts?view=windows-11>

Configuration de la langue : <https://guillaume-cortes.fr/creer-iso-winpe-francais/>

Ajout des drivers PC : <https://dl.dell.com/FOLDER11926959M/1/WinPE10.0-Drivers-A34-PCW4V.cab>

PXE et iPXE pour le boot réseau :

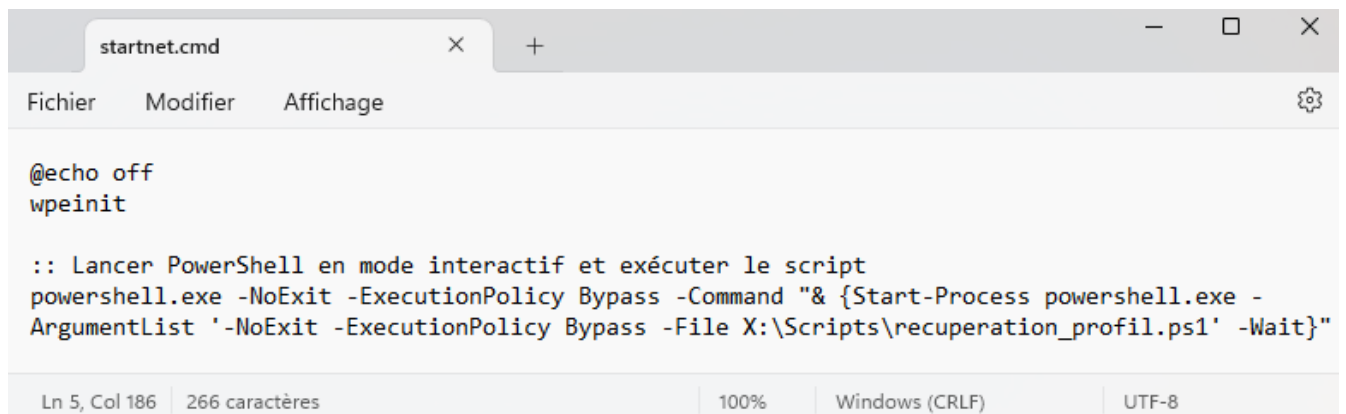
PXE et iPXE : <https://www.it-connect.fr/le-boot-pxe-et-le-boot-ipxe-pour-les-debutants/>

WinPE depuis iPXE : <https://ipxe.org/howto/winpe>

WinPE depuis iXPE : <https://forum.ipxe.org/printthread.php?tid=17332>

10. Annexes

Contenu du fichier de configuration startnet.cmd :



```
@echo off
wpeinit

:: Lancer PowerShell en mode interactif et exécuter le script
powershell.exe -NoExit -ExecutionPolicy Bypass -Command "& {Start-Process powershell.exe -
ArgumentList '-NoExit -ExecutionPolicy Bypass -File X:\Scripts\recuperation_profil.ps1' -Wait}"
```


sgp:recuperation_d_une_sauvegarde_des_profils

Table des matières

- 1) Lancement du script
- 2) Exécution du script
- 3) Problèmes possibles
- ANCIENNE PROCEDURE CI-DESSOUS
- Paramétrages
- Installation d'une nouvelle version linux pour le boot pxe de recupprofil

===== Récupération des profils Windows sur un autre poste =====

1) Lancement du script



Une copie du script se trouve sur `\\servinst.ens-lyon.fr\ghost\migration\script\Script Recup ProfilV2.ps1`

Le principe est de booter une image WinPe installée sur servinst dans le dossier `/var/www/winpe`

- * Booter en iPXE puis choisir l'entrée du menu "Recuperation profil avec winpe"
- * Au démarrage, `startnet.cmd` lance le script "recuperation_profil.ps1" qui se trouve dans `%windir%\System32` du WinPE (généralement X:)

2) Exécution du script

- * Tout d'abord, le script demande le nom de l'ordinateur hôte à déchiffrer : cette variable sera utilisée pour chercher la clé Bitlocker dans l'AD (grâce au crédiel d'opsisuppr) et déverrouiller le lecteur C:
- * Ensuite, entrer le numéro de l'ordinateur de destination où seront réimplantées les données. Cette étape vérifie, puis crée un dossier correspondant sur `Servinst/ghost/migration`
- * Par la suite, il faut sélectionner les profils que l'on souhaite importer sur Servinst

Certains sous-dossiers ou fichiers comme `NTUser.dat` ne sont pas copiés car ils sont obsolètes.

3) Problèmes possibles

- * Le disque dur doit bien être chiffré lors de l'exécution de la procédure, dans le cas contraire, veuillez le chiffrer
- * En cas d'absence de clé BitLocker dans l'Active Directory, il est nécessaire de remonter la clé manuellement depuis l'ordinateur hôte via PowerShell avec ces commandes :

```
$RecoveryKey = ((Get-BitLockerVolume -MountPoint C:).KeyProtector | Where-Object {$_.KeyProtectorType -eq "RecoveryPassword"}).KeyProtector
Backup-BitLockerKeyProtector -MountPoint C: -KeyProtectorId $RecoveryKey
```



Environnement WinPE utilisé pour le configurer :

```
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>dism /Mount-Wim /WimFile:C:\WinPE_amd64_BL\media\sources\boot.wim /index:1 /MountDir:C:\WinPE_amd64_BL\mount

Outil Gestion et maintenance des images de déploiement
Version : 10.0.25398.1

Montage de l'image
[=====100.0%=====]
L'opération a réussi.

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>notepad C:\WinPE_amd64_BL\mount\Windows\System32\Startnet.cmd
C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>notepad C:\WinPE_amd64_BL\mount\Scripts\recuperation_profil.ps1

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>dism /Unmount-Wim /MountDir:C:\WinPE_amd64_BL\mount /Commit

Outil Gestion et maintenance des images de déploiement
Version : 10.0.25398.1

Fichier image : C:\WinPE_amd64_BL\media\sources\boot.wim
Index de l'image : 1
Enregistrement de l'image
[=====100.0%=====]
Démontage de l'image
[=====100.0%=====]
L'opération a réussi.

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools>
```