

Épreuve intégrée – Audit d'un poste Windows via PowerShell (Loïc)

Vous êtes technicien informatique et vous devez collecter des informations essentielles sur un poste Windows afin de les transmettre à un serveur de surveillance du parc.

 niveau

Ce document présente l'énoncé personnalisé de l'épreuve intégrée et précise les informations à collecter pour ce poste.

Contraintes

- Travail en PowerShell, utilisation du pipeline attendue
- Script dans un fichier `.ps1` lisible, structuré et commenté
- `Get-CimInstance` est autorisée et recommandée
- Chaque bloc doit être réalisé dans sa propre fonction
- Aucun accès direct au registre
- Export final d'un objet au format JSON à l'aide de `ConvertTo-Json`
- Respectez **EXACTEMENT** les valeurs qui vous sont demandées

Blocs à réaliser

Informations générales du poste

Objectif : Identifier de manière fiable la machine auditée dans un parc informatique.

A Faire : Récupérer les informations générales permettant d'identifier un poste Windows.

Commandes autorisées / recommandées : `$env:COMPUTERNAME`, `Get-CimInstance Win32_ComputerSystem`

Clé dans le json: `device`

Informations à récupérer

- **Nom de la machine**: Nom local du poste.
- **Constructeur du poste**: Fabricant (ex. Dell, HP, Lenovo).

- **Modèle du poste:** Référence du modèle matériel.
- **RAM totale:** Quantité totale de mémoire vive.
- **Domaine ou groupe de travail:** Contexte d'appartenance réseau du poste.
- **Processeurs logiques:** Nombre de threads visibles par l'OS.
- **Type de système:** Architecture/Type du système (ex. x64).

Traitements à appliquer

- La quantité de RAM doit être affichée en KB.

Processeur (CPU)

Objectif : Décrire les capacités CPU du poste.

A Faire : Récupérer les informations essentielles du processeur.

Commandes autorisées / recommandées : `Get-CimInstance Win32_Processor`

Clé dans le json: `cpu`

Informations à récupérer

Pour tous les objets retournés:

- **Modèle du processeur:** Nom complet du processeur.
- **Nombre de cœurs:** Nombre de cœurs physiques.
- **Nombre de threads:** Nombre de processeurs logiques du CPU.
- **Identifiant CPU:** Identifiant système du CPU.
- **Fabricant CPU:** Fabricant du processeur.
- **Fréquence maximale:** Fréquence max en MHz.

Système d'exploitation

Objectif : Identifier précisément le Windows installé.

A Faire : Récupérer les informations essentielles du système d'exploitation.

Commandes autorisées / recommandées : `Get-CimInstance Win32_OperatingSystem`

Clé dans le json: `os`

Informations à récupérer

- **Nom du système:** Nom complet du système d'exploitation Windows installé.
- **Version:** Version principale du système d'exploitation.

- **Numéro de build:** Numéro de build précis de Windows.
- **Dossier Windows:** Chemin du dossier d'installation de Windows.
- **Architecture OS:** Architecture du système (32-bit ou 64-bit).
- **Numéro de série:** Identifiant ou clé associée au système d'exploitation.

Logiciels installés

Objectif : Réaliser un inventaire applicatif du poste.

A Faire : Lister les logiciels installés et des informations de base.

Commandes autorisées / recommandées : `Get-Package` , `Select-Object` , `Sort-Object`

Clé dans le json: `packages`

Informations à récupérer

Pour tous les objets retournés:

- **Nom du logiciel:** Nom du package/logiciel.
- **Version:** Version installée.
- **Type d'installation:** Origine/gestionnaire (ex. msi, Programs).
- **Statut:** Statut du package (si disponible).
- **Référence interne:** Identifiant interne (si disponible).
- **Source:** Source du package (si disponible).

Traitements à appliquer

- Trier la liste des logiciels par ordre alphabétique
- Ne conserver que les 25 premiers logiciels après tri.

Services Windows

Objectif : Contrôler l'état des services pour diagnostic simple.

A Faire : Lister des services et produire un extrait exploitable.

Commandes autorisées / recommandées : `Get-Service`

Clé dans le json: `services`

Informations à récupérer

Pour tous les objets retournés:

- **Nom du service:** Nom interne du service.

- **Nom affiché:** Nom lisible du service.
- **Statut:** État (Running/Stopped).
- **Peut être arrêté:** Indique si l'arrêt est autorisé.
- **Type de service:** Type (Win32OwnProcess, etc.) si disponible.
- **Pause possible:** Indique si pause/reprise est possible.

Traitements à appliquer

- Ne conserver que les services dont l'état est « en cours d'exécution ».
- Trier les services par nom affiché (DisplayName).
- Ne conserver que les 30 premiers services après tri.

Résultat attendu

Votre script doit produire un **objet** structuré contenant les informations demandées dans les blocs ci-dessus et l'exporter au format JSON.

Votre script **DOIT fonctionner correctement** et fournir les informations demandées.

Présentation orale

Vous devrez être capable d'expliquer :

- comment vous avez identifié les bonnes informations
- pourquoi vous avez choisi certaines commandes
- comment vous avez appliqué les tris / filtres / seuils
- comment vous avez structuré les données
- Vous devrez être en mesure de modifier le script pour ajouter des informations supplémentaires ou corriger des erreurs.