

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

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.
- **Nom DNS:** Nom DNS utilisé sur le réseau.
- **Domaine ou groupe de travail:** Contexte d'appartenance réseau du poste.
- **Présence d'un hyperviseur:** Indique si le système tourne dans un environnement virtualisé.

Traitements à appliquer

- La quantité de RAM doit être affichée en MB.

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.
- **Fréquence maximale:** Fréquence max en MHz.
- **Fabricant CPU:** Fabricant du processeur.
- **Désignation courte:** Nom court (si disponible).

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.
- **Architecture OS:** Architecture du système (32-bit ou 64-bit).
- **Mémoire totale visible:** Quantité totale de mémoire RAM utilisable (en Ko).
- **Dossier système:** Chemin du dossier système principal (ex. System32).

BIOS et carte mère

Objectif : Identifier la plateforme matérielle (firmware).

A Faire : Récupérer des informations BIOS et carte mère utiles à l'inventaire.

Commandes autorisées / recommandées : `Get-CimInstance Win32_BIOS` , `Get-CimInstance Win32_BaseBoard`

Clé dans le json: `firmware`

Informations à récupérer

- **Fabricant du BIOS:** Nom du fabricant du BIOS.
- **Fabricant carte mère:** Nom du fabricant de la carte mère.
- **Modèle carte mère:** Référence (Product) de la carte mère.
- **Version BIOS:** Version installée du BIOS.
- **Numéro de série carte mère:** Identifiant carte mère (si disponible).
- **Date BIOS:** Date de publication du BIOS (si disponible).

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).
- **Pause possible:** Indique si pause/reprise est possible.
- **Peut être arrêté:** Indique si l'arrêt est autorisé.

- **Type de service:** Type (Win32OwnProcess, etc.) si disponible.

Traitements à appliquer

- Ne conserver que les services dont l'état est « arrêté ».
- Trier les services par nom affiché (DisplayName).
- Ne conserver que les 22 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.