# Épreuve intégrée – Audit d'un poste Windows via PowerShell (Manoah Si)

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.
- **Type de système**: Architecture/Type du système (ex. x64).
- **Nom DNS**: Nom DNS utilisé sur le réseau.
- **Domaine ou groupe de travail**: Contexte d'appartenance réseau du poste.

### Traitements à appliquer

- La quantité de RAM doit être affichée en GB.

# 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.
- **Fabricant CPU**: Fabricant du processeur.
- **Fréquence maximale**: Fréquence max en MHz.
- **Socket**: Désignation du socket (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.
- **Mémoire libre**: Quantité de mémoire physique actuellement disponible (en Ko).
- **Organisation**: Organisation déclarée lors de l'installation de Windows.
- **Dernier démarrage**: Date et heure du dernier démarrage du système.

# Cartes graphiques (GPU)

**Objectif** : Inventorier les GPU et leurs pilotes.

**A Faire** : Lister les cartes graphiques détectées sur la machine.

**Commandes autorisées / recommandées** : `Get-CimInstance Win32_VideoController`

**Clé dans le json:** `gpus`

### Informations à récupérer

Pour tous les objets retournés:

- **Nom de la carte graphique**: Nom complet de la carte graphique ou du contrôleur vidéo.
- **Fabricant**: Fabricant ou fournisseur du contrôleur vidéo.
- **Version du pilote**: Version du pilote graphique installé.
- **Mémoire vidéo**: Quantité de mémoire vidéo dédiée (en octets).
- **Architecture vidéo**: Type d'architecture vidéo utilisée par le contrôleur.
- **Résolution horizontale**: Résolution horizontale actuelle de l'écran (en pixels).
- **Mode vidéo**: Description complète du mode vidéo actif (résolution + couleurs).

### Traitements à appliquer

- La quantité de RAM doit être affichée en KB.

# Disques et stockage (volumes)

**Objectif** : Surveiller capacité et espace libre des volumes.

**A Faire** : Lister les volumes/disques logiques et relever les informations clés.

**Commandes autorisées / recommandées** : `Get-CimInstance Win32_LogicalDisk`

**Clé dans le json:** `volumes`

### Informations à récupérer

Pour tous les objets retournés:

- **Lettre/identifiant du volume**: Ex. C: , D: ...

- **Système de fichiers**: Ex. NTFS, FAT32.
- **Espace libre (octets)**: Espace libre disponible sur le volume.
- **Type de lecteur**: Type du volume (local, réseau, etc.).
- **Chemin réseau**: Nom du partage si lecteur réseau.
- **Taille totale (octets)**: Capacité totale du volume.

### Traitements à appliquer

- Ne conserver que les disques locaux (indice: leur DriveType = 3).
- Trier par espace libre croissant et ne conserver que les 2 premiers volumes.

# 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.