

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

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.
- **Processeurs logiques:** Nombre de threads visibles par l'OS.
- **Utilisateur connecté:** Compte utilisateur actif au moment de l'exécution.
- **Présence d'un hyperviseur:** Indique si le système tourne dans un environnement virtualisé.

### 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.
- **Socket:** Désignation du socket (si disponible).
- **Identifiant CPU:** Identifiant système du CPU.
- **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.
- **Numéro de série:** Identifiant ou clé associée au système d'exploitation.
- **Dossier Windows:** Chemin du dossier d'installation de Windows.
- **Langues MUI:** Liste des langues d'interface utilisateur installées sur le système.

## Adaptateurs réseau

**Objectif :** Inventorier l'état des interfaces réseau.

**A Faire :** Lister les adaptateurs réseau et leur état.

**Commandes autorisées / recommandées :** `Get-NetAdapter`

**Clé dans le json:** `networkAdapters`

### Informations à récupérer

Pour tous les objets retournés:

- **Nom de l'interface:** Nom logique de l'interface réseau.
- **Description:** Description matérielle complète de la carte réseau.
- **Statut:** État actuel de l'interface réseau (Up, Down, Disabled).
- **Vitesse du lien:** Vitesse de connexion négociée de l'interface réseau.
- **Version du pilote:** Version du pilote de la carte réseau.
- **MTU:** Taille maximale des trames réseau (MTU).

### Traitements à appliquer

- Ne conserver que les interfaces dont l'état est « Up ».
- Ne conserver que les interfaces virtuelles.
- Trier les interfaces par vitesse de connexion décroissante.

## Dossier Téléchargements (fichiers)

**Objectif :** Repérer des fichiers lourds / suspects dans Téléchargements.

**A Faire :** Lister des fichiers du dossier Téléchargements avec filtrage, tri et Top N.

**Commandes autorisées / recommandées :** `Get-ChildItem`

**Clé dans le json:** `downloadsFiles`

### Informations à récupérer

Pour tous les objets retournés:

- **Nom du fichier:** Nom du fichier.
- **Taille (octets):** Taille du fichier en octets.
- **Dernière modification:** Date/heure de dernière modification.
- **Date de création:** Date/heure de création.
- **Chemin complet:** Chemin complet du fichier.
- **Extension:** Extension du fichier.

### Traitements à appliquer

- Ne conserver que les fichiers dont la taille est  $\geq 90000000$  bytes.
- Trier les fichiers par ordre alphabétique (Nom)..
- Ne conserver que les 9 premiers fichiers 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.