BTS SIO — Option SISR — Fiche de Réalisation Professionnelle

Supervision Avancée via WMI (Windows Management Instrumentation)

Interrogation CIM/WQL depuis Zabbix — SRV-AD-01 (10.11.3.18 — Windows Server)

Zabbix Agent | Clé wmi.get | Classe Win32_OperatingSystem | Namespace Root\CIMv2

EN-TÊTE — TABLEAU RÉCAPITULATIF	
Champ	**Valeur**
Candidat	BTS SIO option SISR — Session 2026
Entreprise	SIO SISR
Objet	Supervision avancée Windows via requêtes WMI/WQL intégrées dans Zabbix
Cible supervisée	SRV-AD-01 — Windows Server — 10.11.3.18
Outil de supervision	Zabbix 6.x / 7.x — Clé wmi.get — Namespace Root\CIMv2
Référentiel SISR	Compétences : Gestion des indicateurs de performance — Supervision système
Date	Mai 2026
Fiche n°	E6 — Supervision et Maintenance

I. INTRODUCTION TECHNIQUE — WMI, CIM et WQL

WMI (Windows Management Instrumentation) est le framework de gestion système de Microsoft, basé sur le standard DMTF CIM. Il expose une base hiérarchique organisée en namespaces (ex : **Root\CIMv2**), contenant des classes représentant chaque composant du système (matériel, OS, processus, services). Zabbix exploite WMI via l'agent Windows pour interroger ces classes avec des requêtes **WQL (WMI Query Language)**, syntaxe proche du SQL.

Tableau comparatif — Agent Zabbix classique vs WMI (wmi.get)

Critère / Agent Zabbix standard / WMI — wmi.get (WQL)		
Critère	**Agent Zabbix (clés standard)**	**WMI / wmi.get (WQL)**
Données collectées	Métriques prédéfinies (CPU, RAM, disque, réseau)	Toute donnée CIM : BIOS, matériel, OS, licences, services...
Granularité	Limitée aux clés du template	Fine — accès direct aux propriétés de chaque classe Win32_*
Requête personnalisée	Non (clé fixe, ex: system.cpu.util)	Oui — WQL libre : SELECT Caption FROM Win32_OperatingSystem
Dépendances cible	Agent Zabbix installé et configuré	Agent + Winmgmt actif + droits namespace Root\CIMv2
Cas d'usage SISR	Supervision opérationnelle standard (MCO)	Audit matériel, conformité OS, supervision applicative avancée
Complexité	Faible — plug & play avec templates officiels	Moyenne — droits WMI, pare-feu RPC, maîtrise WQL

II. PRÉPARATION DE LA CIBLE — SRV-AD-01 (10.11.3.18)

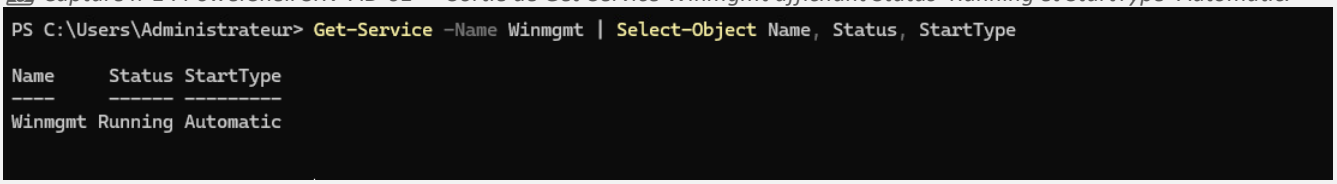
Étape 1 — Vérification et activation du service Winmgmt

Navigation : SRV-AD-01 — PowerShell (Administrateur)

```
# Vérifier le statut du service WMI
Get-Service -Name Winmgmt
# Résultat attendu : Status = Running

# Si arrêté, démarrer et configurer en automatique :
Start-Service -Name Winmgmt
Set-Service -Name Winmgmt -StartupType Automatic

# Vérification post-activation :
Get-Service -Name Winmgmt | Select-Object Name, Status, StartType
# Résultat attendu : Name=Winmgmt  Status=Running  StartType=Automatic
```

Service Winmgmt : Status = Running | StartType = Automatic — WMI opérationnel sur SRV-AD-01.

Capture n°1 : PowerShell SRV-AD-01 — Sortie de Get-Service Winmgmt affichant Status=Running et StartType=Automatic.



Étape 2 — Configuration du pare-feu Windows (RPC et WMI)

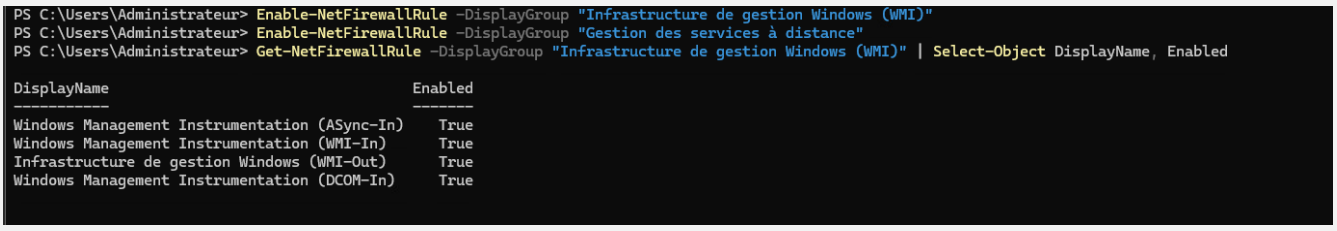
Navigation : SRV-AD-01 — PowerShell (Administrateur)

```
# Autoriser WMI dans le pare-feu Windows (groupe de règles dédié)
Enable-NetFirewallRule -DisplayGroup "Windows Management Instrumentation (WMI)"

# Autoriser DCOM/RPC (requis pour les appels WMI distants)
Enable-NetFirewallRule -DisplayGroup "Remote Administration"

# Vérification des règles activées :
Get-NetFirewallRule -DisplayGroup "Windows Management Instrumentation (WMI)" |
    Select-Object DisplayName, Enabled, Direction
# Résultat attendu : Enabled = True pour toutes les règles WMI-In

# Alternative manuelle : wf.msc > Inbound Rules > WMI-In > Enable
```

⚠ SÉCURITÉ : En production, restreindre les règles WMI à l'IP de SRV-ZABBIX uniquement (RemoteAddress = 10.11.4.x) et non à Any. Cela réduit la surface d'attaque, conformément aux recommandations ANSSI sur la sécurisation des interfaces d'administration.

Règles pare-feu WMI Inbound activées — WMI accessible depuis le réseau interne.

Capture n°2 : PowerShell — Sortie de Get-NetFirewallRule affichant les règles WMI avec Enabled=True et Direction=Inbound.



Étape 3 — Gestion des droits sur le namespace Root\CIMv2

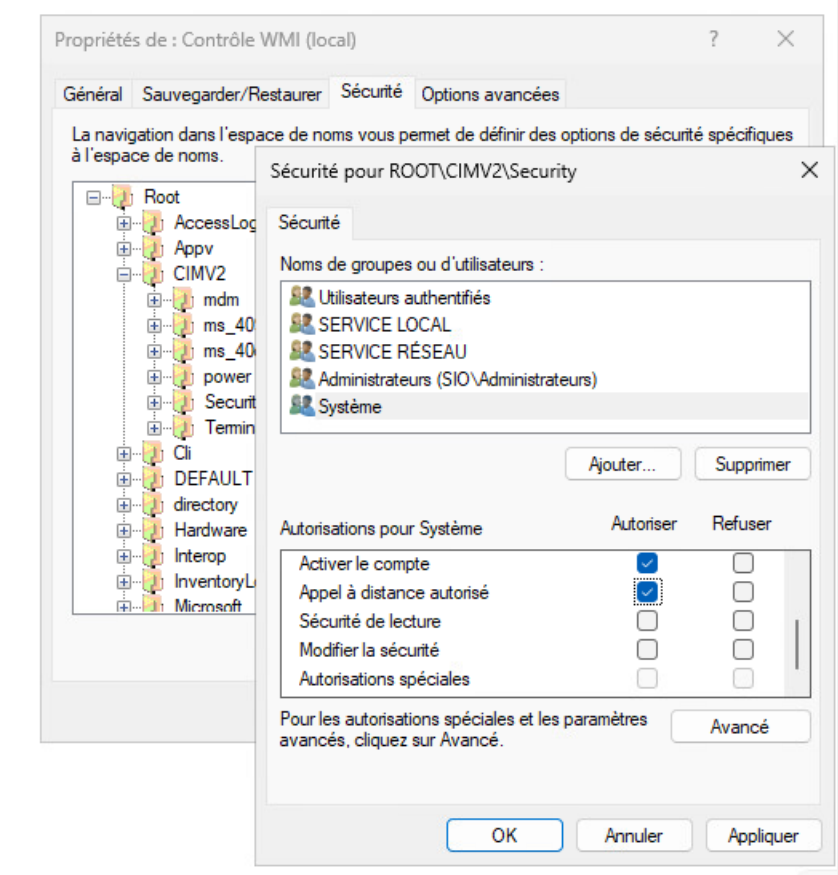
Navigation : SRV-AD-01 — wmimgmt.msc (Console WMI) ou PowerShell (Administrateur)

```
# Méthode graphique (wmimgmt.msc) :
# wmimgmt.msc > Clic droit "WMI Control (Local)" > Properties
# > Onglet Security > Root > CIMv2 > [Security]
# > Ajouter le compte de service Zabbix (ex : SYSTEM ou zabbix-svc)
# > Droits requis : Enable Account + Remote Enable

# Test WQL local depuis PowerShell (validation côté OS) :
Get-WmiObject -Namespace "root/cimv2" `
    -Query "SELECT Caption FROM Win32_OperatingSystem" |
    Select-Object Caption
# Résultat attendu : Caption = Windows Server 20xx Standard/Datacenter

# Vérification que l'agent Zabbix tourne bien en compte SYSTEM :
Get-WmiObject Win32_Service -Filter "Name='Zabbix Agent'" |
    Select-Object Name, StartName, State
# Résultat attendu : StartName = LocalSystem
```

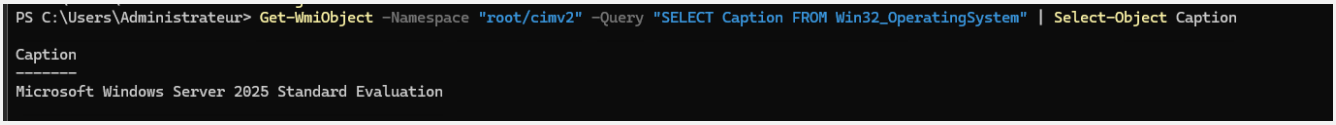
⚠ PRÉREQUIS CRITIQUE : Le compte utilisé par l'agent Zabbix (SYSTEM ou compte dédié) doit avoir les droits "Enable Account" ET "Remote Enable" sur Root\CIMv2. Sans ces droits, wmi.get retournera ZBX_NOTSUPPORTED.

☑ Test PowerShell local réussi — Win32_OperatingSystem.Caption retourne le nom de l'OS. Les droits namespace sont correctement attribués.

📷 Capture n°3 : wmimgmt.msc — Fenêtre Security du namespace Root\CIMv2 avec droits "Enable Account" + "Remote Enable" cochés pour le compte Zabbix.



Capture n°4 : PowerShell SRV-AD-01 — Sortie de Get-WmiObject avec requête WQL retournant Caption=Windows Server ... : Get-WmiObject -Namespace "root/cimv2" -Query "SELECT Caption FROM Win32_OperatingSystem" | Select-Object Caption



III. MISE EN ŒUVRE DANS ZABBIX — Création de l'Item wmi.get

La clé **wmi.get[namespace,query]** est une clé native de l'agent Zabbix Windows permettant d'exécuter une requête WQL sur la cible locale et de retourner la valeur de la **première propriété du premier objet** trouvé. Elle est exécutée côté agent (sur SRV-AD-01), sans accès réseau WMI direct depuis SRV-ZABBIX.

Étape 4 — Création de l'Item Zabbix (type Zabbix Agent)

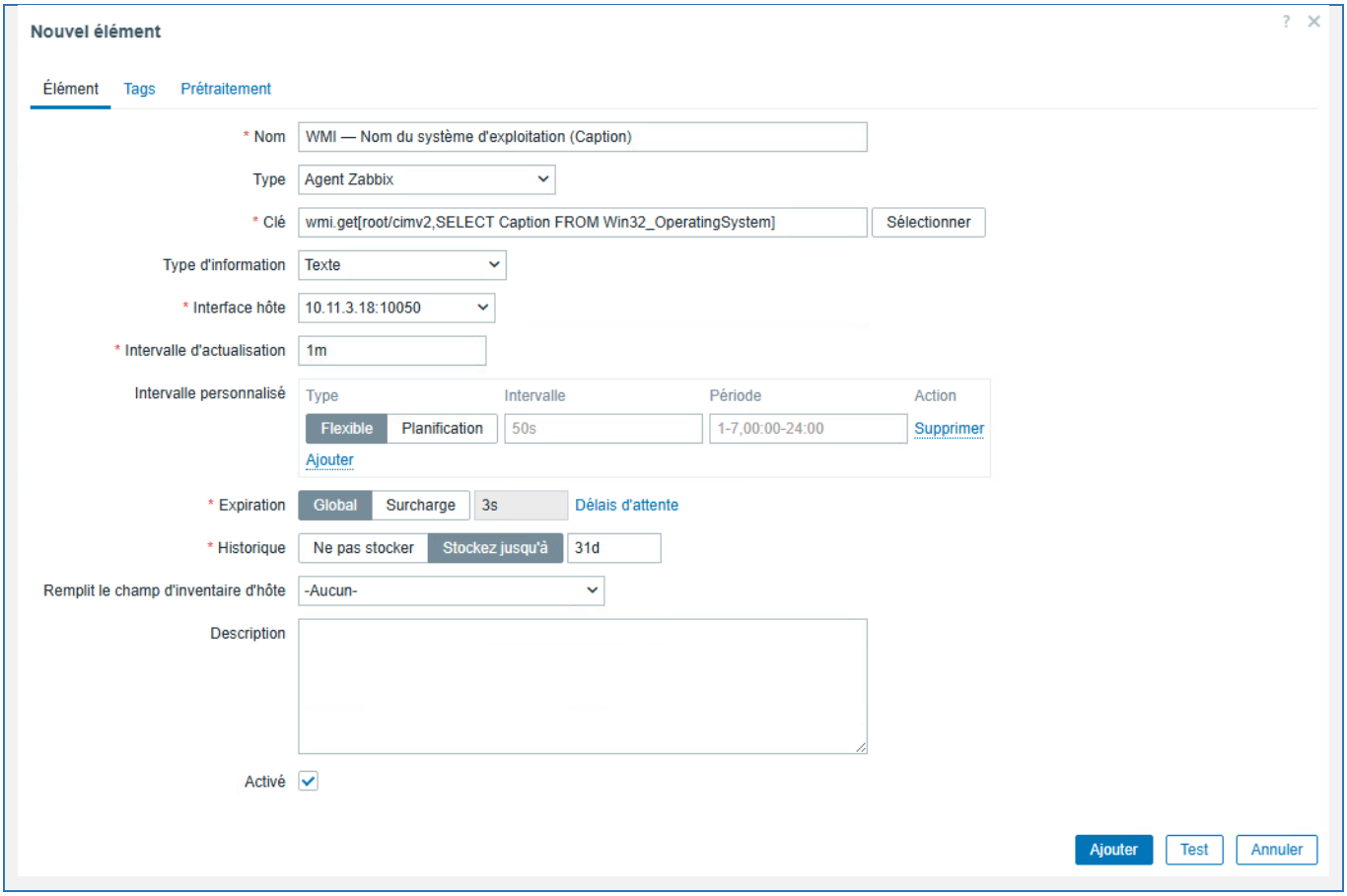
Navigation : Interface Web Zabbix > Configuration > Hosts > SRV-AD-01 > Items > [Create item]

```
# Paramètres à renseigner dans le formulaire Create item :

Name               : WMI – Nom du système d'exploitation (Caption)
Type               : Zabbix agent
Key                : wmi.get[root/cimv2,SELECT Caption FROM Win32_OperatingSystem]
Type of information: Text
Update interval    : 1h  (3600s – donnée statique, polling fréquent non requis)
History storage    : 90 days
Description        : Récupère le nom exact de l'OS via WMI – classe Win32_OperatingSystem
Applications       : WMI / Supervision OS

# ──────────────────────────────────────────────────────────
# Syntaxe générale de la clé wmi.get :
# wmi.get[<namespace>,<requête WQL>]
# ──────────────────────────────────────────────────────────

# Autres clés WMI exploitables sur SRV-AD-01 :
wmi.get[root/cimv2,SELECT SerialNumber FROM Win32_BIOS]
    → Numéro de série du BIOS (audit inventaire)

wmi.get[root/cimv2,SELECT TotalPhysicalMemory FROM Win32_ComputerSystem]
    → RAM totale installée (en octets)

wmi.get[root/cimv2,SELECT Version FROM Win32_OperatingSystem]
    → Version exacte du kernel Windows (ex: 10.0.20348)

wmi.get[root/cimv2,SELECT LastBootUpTime FROM Win32_OperatingSystem]
    → Dernier démarrage du serveur (timestamp CIM)
```

*⚠ SYNTAXE CRITIQUE : wmi.get retourne UNIQUEMENT la valeur de la première propriété du premier objet retourné. Toujours cibler une propriété précise (SELECT Caption, SELECT SerialNumber...). N'utiliser jamais SELECT * — seule la première colonne serait capturée, de façon imprévisible.*

☑ Item créé et sauvegardé — statut Enabled. Zabbix transmettra la clé wmi.get à l'agent de SRV-AD-01 lors du prochain cycle de polling.

Capture n°5 : Interface Zabbix — Formulaire Create Item avec champs Name, Type (Zabbix agent), Key wmi.get[root/cimv2,SELECT Caption FROM Win32_OperatingSystem] et Type of information (Text) remplis.
→



IV. TESTS ET VALIDATION — Vérification de la Remontée WMI dans Zabbix

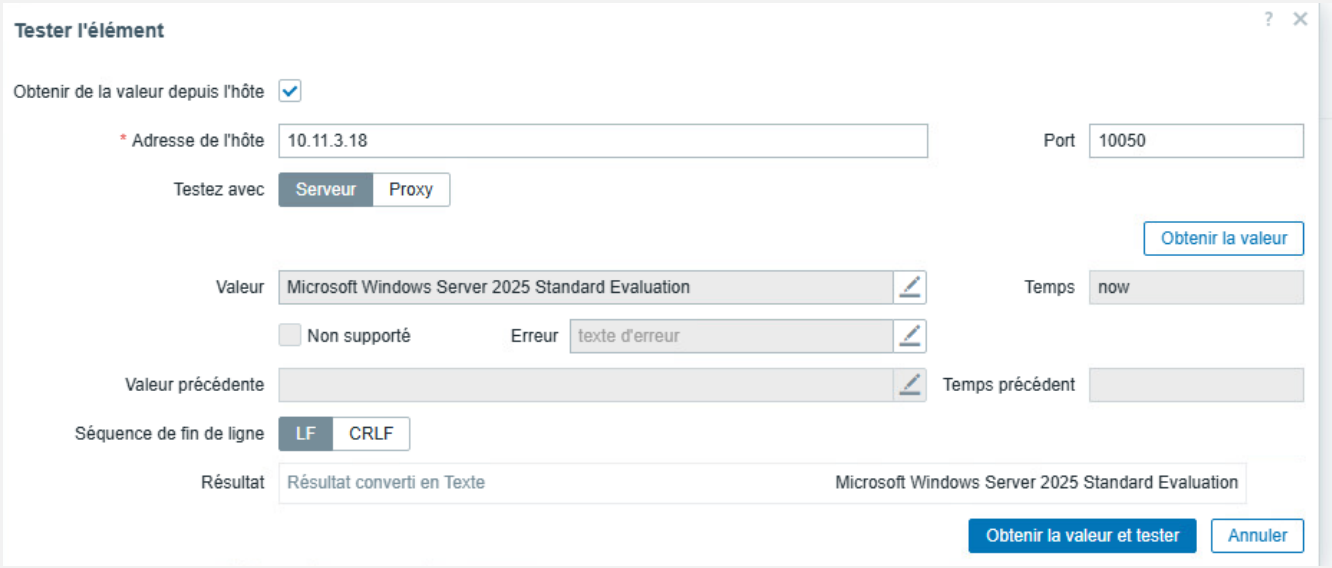
Étape 5 — Test immédiat via [Get value and test]

Navigation : Configuration > Hosts > SRV-AD-01 > Items > [item WMI] > [Test]

```
# Procédure de test immédiat :
1. Configuration > Hosts > SRV-AD-01 > Items
2. Localiser l'item "WMI — Nom du système d'exploitation (Caption)"
3. Cliquer sur [Test] en bout de ligne
4. Dans la fenêtre : cliquer sur [Get value and test]

# Résultat attendu dans le champ "Value" :
Value: Windows Server 2022 Standard
       (ou Windows Server 2025 selon la version installée sur SRV-AD-01)

# ─── Gestion des erreurs ──────────────────────────────────
ZBX_NOTSUPPORTED: WMI is not available
    → Vérifier : service Winmgmt Running | agent Zabbix en SYSTEM

ZBX_NOTSUPPORTED: Cannot obtain WMI information
    → Vérifier la syntaxe WQL et le nom exact de la classe Win32_OperatingSystem

ZBX_NOTSUPPORTED: Failed to connect to WMI namespace
    → Vérifier les droits "Enable Account" + "Remote Enable" sur Root\CIMv2
```

Test Zabbix réussi — Value : "Windows Server 202x Standard" retourné sans erreur depuis wmi.get.

Capture n°6 : Interface Zabbix — Fenêtre Test de l'item WMI affichant Value = "Windows Server 202x Standard" après [Get value and test].



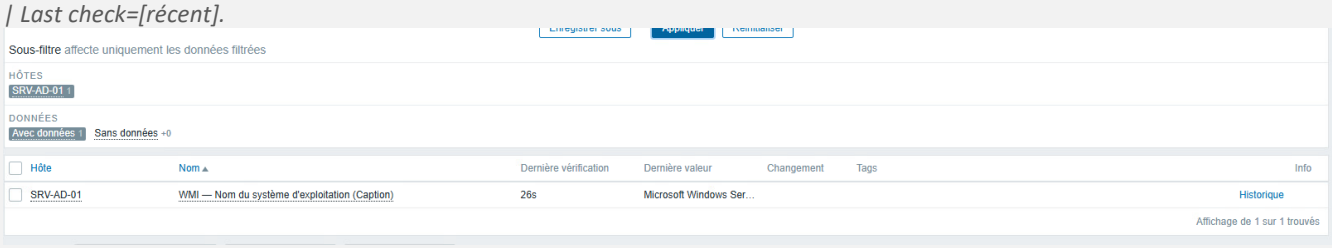
Étape 6 — Vérification dans Monitoring > Latest data

Navigation : Monitoring > Latest data > Filtre : Host = SRV-AD-01

```
# Vérification dans Latest data :
1. Monitoring > Latest data
2. Filtre : Host = SRV-AD-01  |  Name = WMI (ou Caption)
3. Cliquer sur [Apply]

# Colonnes à contrôler :
Host       : SRV-AD-01
Name       : WMI — Nom du système d'exploitation (Caption)
Last value : Windows Server 202x Standard
Last check : [horodatage récent — inférieur à 1h]

# Si Last check est ancien ou Last value vide :
→ Configuration > Hosts > SRV-AD-01 > Items
   → Sélectionner l'item > [Execute now]
   → Attendre 30s et rafraîchir Monitoring > Latest data
```

Latest data : valeur "Windows Server 202x Standard" présente, Last check récent. Supervision WMI opérationnelle et intégrée dans le flux de collecte Zabbix.

Capture n°7 : Monitoring > Latest data — Ligne de l'item WMI : Host=SRV-AD-01 | Last value=Windows Server 202x Standard | Last check=[récent].



Étape 7 — Diagnostic avancé (zabbix_get + logs agent)

Navigation : SRV-ZABBIX — Terminal + SRV-AD-01 — PowerShell (Administrateur)

```
# Test depuis SRV-ZABBIX via zabbix_get (outil de diagnostic externe) :
zabbix_get -s 10.11.3.18 -p 10050 \
    -k "wmi.get[root/cimv2,SELECT Caption FROM Win32_OperatingSystem]"
# Résultat attendu : Windows Server 202x Standard

# Consulter les logs de l'agent Zabbix sur SRV-AD-01 :
Get-Content "C:\Program Files\Zabbix Agent\zabbix_agentd.log" -Tail 50 |
    Select-String "wmi|WMI|error|ERROR"

# Test WQL direct depuis PowerShell SRV-AD-01 (validation OS) :
Get-WmiObject -Namespace "root/cimv2" `
    -Query "SELECT Caption FROM Win32_OperatingSystem" |
    Select-Object Caption
# Résultat attendu : Caption = Windows Server 202x Standard
```

⚠ POINT JURY : La clé wmi.get est exécutée LOCALEMENT par l'agent Zabbix sur SRV-AD-01 (pas depuis SRV-ZABBIX via le réseau). L'agent interroge WMI en SYSTEM local — ce n'est pas du WMI distant (DCOM). Cela simplifie la configuration réseau mais requiert impérativement que l'agent Zabbix soit installé sur chaque cible Windows.

V. BILAN ET ARGUMENTAIRE — Référentiel SISR & Compétences E6			
#	**Action réalisée**	**Résultat observé**	**Statut**
1	Vérification et activation du service Winmgmt	Status=Running, StartType=Automatic — PowerShell confirmé	☑ OK
2	Ouverture règles pare-feu WMI Inbound	Get-NetFirewallRule : Enabled=True pour toutes les règles WMI-In	☑ OK
3	Droits Root\CIMv2 attribués au compte agent Zabbix	Enable Account + Remote Enable configurés via wmimgmt.msc	☑ OK
4	Création de l'item wmi.get dans Zabbix (SRV-AD-01)	Item Zabbix Agent — clé wmi.get[root/cimv2,SELECT Caption...]	☑ OK
5	Test immédiat via [Get value and test]	Value = "Windows Server 202x Standard" — aucune erreur	☑ OK
6	Vérification dans Monitoring > Latest data	Donnée présente, Last check récent — intégration confirmée	☑ OK
7	Diagnostic via zabbix_get et logs agent	Procédure de dépannage documentée et validée	☑ OK

Lien avec le Référentiel BTS SIO SISR — Compétences E6

▶ **Gestion des indicateurs de performance** — La clé wmi.get collecte des métriques non exposées par les clés standard, répondant directement à cette compétence référentiel.

▶ **MCO (Maintien en Conditions Opérationnelles)** — La supervision WMI couvre les composants bas niveau (BIOS, matériel, licences OS) invisibles avec l'agent standard.

▶ **Gestion de la sécurité** — La configuration fine des droits WMI (namespace, compte de service) démontre la maîtrise du principe du moindre privilège.

▶ **Documentation et traçabilité** — Cette procédure constitue un livrable de mission SISR, directement intégrable au dossier E6.

Conclusion technique — Mots-clés BTS SIO SISR

La mise en œuvre de la supervision WMI via la clé **wmi.get** dans Zabbix étend les capacités de collecte bien au-delà des templates standards. En interrogeant directement la base **CIM (Common Information Model)** via des

requêtes **WQL**, l'administrateur accède à des données système profondes (BIOS, OS, matériel) indisponibles avec les clés standard — renforçant ainsi le dispositif de **supervision proactive** et du **MCO (Maintien en Conditions Opérationnelles)**.

🎯 **Mots-clés jury :** *WMI • CIM • WQL • wmi.get • Win32_OperatingSystem • Root\CIMv2 • Supervision proactive • MCO • Moindre privilège • Indicateurs de performance • Zabbix Agent*