Détecter les Attaques Active Directory avec Wazuh : Guide Complet

> Détecter les attaques Active Directory avec Wazuh : règles pour DCSync, Kerberoasting, Pass-the-Hash, Golden Ticket. Intégration Windows Event Forwarding.

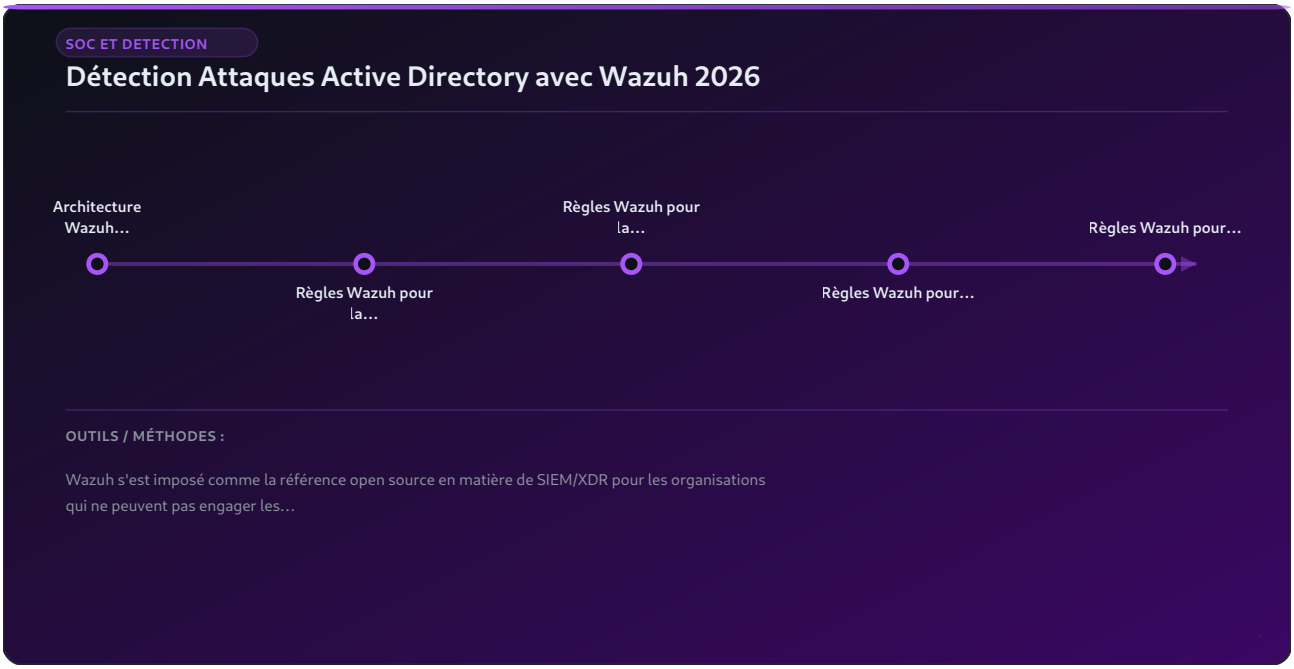
Wazuh s'est imposé comme la référence open source en matière de SIEM/XDR pour les organisations qui ne peuvent pas engager les budgets de Splunk, Microsoft Sentinel ou IBM QRadar. Pour les équipes SOC françaises en charge de la surveillance d'Active Directory, Wazuh offre une couverture remarquable des attaques Kerberos et NTLM, à condition de configurer correctement la collecte des événements Windows sur les contrôleurs de domaine et de développer des règles de détection précises. L'Active Directory reste la cible n°1 des attaquants qui cherchent à élever leurs privilèges et à se déplacer latéralement dans le réseau : DCSync permet d'extraire tous les hashes NTLM en simulant un DC, Kerberoasting cible les comptes de service, Pass-the-Hash exploite la réutilisation des hashes NTLM, et les Golden Tickets permettent de forger des authentifications Kerberos indétectables sans une surveillance précise. Ce guide couvre l'architecture Wazuh pour surveiller un Active Directory multi-DC, la configuration des canaux d'événements Windows critiques, et les règles XML de détection pour chacune des principales techniques d'attaque référencées dans MITRE ATT&CK. Il inclut des retours d'expérience de SOC PME françaises ayant déployé Wazuh avec MISP pour la corrélation des IoC ANSSI.

Points clés à retenir

• Le cloud shift impose une adaptation profonde des pratiques de sécurité traditionnelles

• La gestion des identités et des accès est le périmètre de sécurité du cloud moderne

• Les misconfigurations restent la première cause de compromission dans les environnements cloud



Architecture Wazuh pour la surveillance Active Directory

Une surveillance efficace d'Active Directory avec Wazuh repose sur trois composants : les agents Wazuh déployés sur chaque contrôleur de domaine, le Windows Event Forwarding (WEF) comme canal de collecte centralisé, et le serveur Wazuh (manager) qui héberge les règles de corrélation.

⚡ **Retour terrain**

Dans les SOC que j'accompagne, le principal problème n'est pas le manque d'alertes mais l'excès : 'alert fatigue' systématique, analysts qui désactivent des règles jugées trop

bruyantes sans documentation, et faux positifs récurrents qui masquent les vrais incidents. J'ai développé un tableau de bord simple — 3 métriques : taux de faux positifs par règle, MTTD (mean time to detect) sur incidents réels, et ratio alertes/analyst — qui permet en une réunion hebdomadaire de 30 minutes d'identifier les règles à tuner en priorité.

Déploiement des agents Wazuh sur les DC

Chaque contrôleur de domaine doit disposer d'un agent Wazuh local. L'agent en mode centralisé (forwarding uniquement vers un Windows Event Collector) ne suffit pas : des règles critiques comme la détection de modifications de NTDS.dit ou les événements Sysmon nécessitent un agent local capable d'exécuter des commandes et des scripts de réponse automatique (active response).

```
 # Déploiement de l'agent Wazuh sur un DC Windows Server 2022
# PowerShell (exécuté depuis le DC)
Invoke-WebRequest -Uri "https://packages.wazuh.com/4.x/windows/wazuh-agent-4.x.x.msi" -OutFile "w
Start-Process msiexec.exe -ArgumentList "/i wazuh-agent.msi WAZUH_MANAGER=wazuh-server.domaine.lo

# Démarrer l'agent
Start-Service WazuhSvc
Set-Service WazuhSvc -StartupType Automatic

# Vérifier l'enregistrement
Get-Service WazuhSvc | Select-Object Status, StartType
```

Windows Event Forwarding (WEF) — Architecture recommandée

Le WEF permet de centraliser les événements de plusieurs DC vers un Windows Event Collector (WEC) dédié. Cette architecture est recommandée pour les environnements avec plus de 5 DC, car elle réduit la charge sur chaque agent Wazuh tout en centralisant les logs pour corrélation.

```
 # Sur le Windows Event Collector (WEC) — activer le service
wecutil qc

# Créer un abonnement WEF vers les DC (depuis le WEC)
wecutil cs subscription-ad-security.xml

# Vérifier les sources connectées
wecutil gs "AD-Security-Subscription"
wecutil gr "AD-Security-Subscription"
```

Configuration des canaux Windows Events sur les DC

La collecte des bons événements est la fondation de toute détection AD. Par défaut, Windows n'active pas le niveau de verbosité nécessaire. Une politique d'audit avancée doit être configurée via GPO sur tous les DC.

```
 # Configurer l'audit avancé sur les DC (via PowerShell ou GPO)
auditpol /set /subcategory:"Directory Service Access" /success:enable /failure:enable
auditpol /set /subcategory:"Directory Service Changes" /success:enable /failure:enable
auditpol /set /subcategory:"Kerberos Service Ticket Operations" /success:enable /failure:enable
auditpol /set /subcategory:"Kerberos Authentication Service" /success:enable /failure:enable
auditpol /set /subcategory:"Logon" /success:enable /failure:enable
auditpol /set /subcategory:"Account Logon" /success:enable /failure:enable
auditpol /set /subcategory:"Security Group Management" /success:enable /failure:enable
auditpol /set /subcategory:"Computer Account Management" /success:enable /failure:enable

# Vérifier la configuration d'audit
auditpol /get /category:* | Where-Object {$_ -match "Success and Failure|Success|Failure"}
```

Configurez le collecteur Wazuh pour ingérer les canaux suivants sur chaque DC, dans le fichier de configuration de l'agent `ossec.conf` :

```
 <!-- ossec.conf sur les DC — canaux à collecter -->
<ossec_config>
  <localfile>
    <location>Security</location>
    <log_format>eventchannel</log_format>
    <query>Event/System[EventID != 4688 and EventID != 4624]</query>
  </localfile>

  <localfile>
    <location>System</location>
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>Directory Service</location>
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>DNS Server</location>
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>Microsoft-Windows-Sysmon/Operational</location>
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>Microsoft-Windows-PowerShell/Operational</location>
    <log_format>eventchannel</log_format>
  </localfile>
</ossec_config>
```

Règles Wazuh pour la détection DCSync

DCSync est une technique d'extraction des hashes NTLM qui simule un contrôleur de domaine demandant une réplication. Elle génère des événements 4662 avec des GUID de droits de réplication spécifiques. Pour aller plus loin sur les mécanismes d'attaque et de défense, consultez notre article détaillé sur DCSync — attaque et défense.

```
 <!-- Règle DCSync - Event 4662 avec GUID réplication -->
<rule id="100200" level="14">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4662$</field>
  <field name="win.eventdata.properties">{1131f6aa-9c07-11d1-f79f-00c04fc2dcd2}|{1131f6ab-9c07-11
  <description>DCSync Attack Detected - Replication rights exercised</description>
  <mitre><id>T1003.006</id></mitre>
</rule>

<!-- Règle DCSync enrichie - corrélation avec comptes non-DC -->
<rule id="100201" level="15" frequency="2" timeframe="60">
  <if_matched_sid>100200</if_matched_sid>
  <same_field>win.eventdata.subjectUserName</same_field>
  <description>DCSync Attack Confirmed - Multiple replication requests from same account (not a D
  <mitre><id>T1003.006</id></mitre>
  <group>attack,credential_access</group>
</rule>

<!-- Event 4661 - accès objet sensible (précurseur DCSync) -->
<rule id="100202" level="10">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4661$</field>
  <field name="win.eventdata.objectType">SAM_DOMAIN</field>
  <field name="win.eventdata.accessMask">0x100</field>
  <description>Potential DCSync Precursor - SAM Domain object access</description>
  <mitre><id>T1003.006</id></mitre>
</rule>
```

Règles Wazuh pour la détection Kerberoasting

Le Kerberoasting exploite le protocole Kerberos pour extraire les hashes des comptes de service disposant d'un SPN. Il se manifeste par des demandes de tickets TGS avec chiffrement RC4 (type 0x17) qui peuvent être craqués hors ligne. Notre article sur le Kerberoasting — attaque et défense détaille les mécanismes d'exploitation.

```
 <!-- Règle Kerberoasting - RC4 TGS request -->
<rule id="100210" level="12">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4769$</field>
  <field name="win.eventdata.ticketEncryptionType">^0x17$</field>
  <field name="win.eventdata.ticketOptions">^0x40810000$</field>
  <description>Kerberoasting Attack - RC4 TGS requested</description>
  <mitre><id>T1558.003</id></mitre>
</rule>

<!-- Kerberoasting en masse - volume anormal de TGS RC4 -->
<rule id="100211" level="14" frequency="5" timeframe="60">
  <if_matched_sid>100210</if_matched_sid>
  <same_field>win.eventdata.clientAddress</same_field>
  <description>Kerberoasting Campaign - Multiple RC4 TGS requests from same source</description>
  <mitre><id>T1558.003</id></mitre>
  <group>attack,credential_access</group>
</rule>

<!-- AS-REP Roasting - comptes sans pré-authentification Kerberos -->
<rule id="100212" level="10">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4768$</field>
  <field name="win.eventdata.preAuthType">^0$</field>
  <description>AS-REP Roasting - Kerberos pre-authentication not required</description>
  <mitre><id>T1558.004</id></mitre>
</rule>
```

Règles Wazuh pour Pass-the-Hash

Pass-the-Hash exploite les hashes NTLM pour s'authentifier sans connaître le mot de passe en clair. La détection repose sur la corrélation d'événements 4624 (Logon Type 3 = réseau) avec le package d'authentification NTLM sur des comptes sensibles. Consultez notre article sur Pass-the-Hash et les attaques NTLM sur Active Directory pour le contexte complet.

```
 <!-- Pass-the-Hash - NTLM Network logon compte sensible -->
<rule id="100220" level="12">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4624$</field>
  <field name="win.eventdata.logonType">^3$</field>
  <field name="win.eventdata.authenticationPackageName">^NTLM$</field>
  <list field="win.eventdata.targetUserName" lookup="match_key">etc/lists/sensitive-accounts</lis
  <description>Pass-the-Hash Suspected - Privileged NTLM network logon</description>
  <mitre><id>T1550.002</id></mitre>
</rule>

<!-- PtH - NTLM depuis source inhabituelle sur DC -->
<rule id="100221" level="13">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4624$</field>
  <field name="win.eventdata.logonType">^3$</field>
  <field name="win.eventdata.authenticationPackageName">^NTLM$</field>
  <field name="win.eventdata.workstationName">\.
```

Règles Wazuh pour Golden Ticket

La détection des Golden Tickets est difficile car Kerberos est conçu pour valider les tickets sans requête DC. Les indicateurs les plus fiables sont les anomalies dans les événements 4768 (TGT request) : durée de vie du ticket anormale, type de chiffrement inhabituel, ou demande TGT pour un compte administrateur depuis une source inattendue.

```
 <!-- Golden Ticket - TGS avec options suspectes et RC4 -->
<rule id="100230" level="13">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4769$</field>
  <field name="win.eventdata.ticketOptions">^0x40810010$</field>
  <field name="win.eventdata.ticketEncryptionType">^0x17$</field>
  <description>Potential Golden Ticket - Forged TGS with RC4 and anomalous options</description>
  <mitre><id>T1558.001</id></mitre>
</rule>

<!-- Golden Ticket - compte admin TGS request RC4 -->
<rule id="100231" level="14">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4769$</field>
  <list field="win.eventdata.accountName" lookup="match_key">etc/lists/sensitive-accounts</list>
  <field name="win.eventdata.ticketEncryptionType">^0x17$</field>
  <description>Golden Ticket Suspected - Admin account TGS request with RC4 encryption</descripti
  <mitre><id>T1558.001</id></mitre>
  <group>attack,persistence</group>
</rule>
```

Règles Wazuh pour la reconnaissance LDAP (BloodHound)

BloodHound effectue une reconnaissance LDAP massive pour cartographier les chemins d'attaque AD. L'Event 1644 du journal Directory Service enregistre les requêtes LDAP coûteuses. Un pic de requêtes LDAP non indexées depuis une workstation est un signal fort d'activité BloodHound. Pour le contexte des chemins d'attaque, voir notre article sur BloodHound et la cartographie des chemins d'attaque AD.

```
 <!-- LDAP Recon - Event 1644 (requêtes LDAP non optimisées) -->
<rule id="100240" level="10">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^1644$</field>
  <field name="win.system.channel">Directory Service</field>
  <description>LDAP Expensive Query - Potential AD reconnaissance</description>
  <mitre><id>T1018</id></mitre>
</rule>

<!-- BloodHound/LDAP recon en masse - volume anormal -->
<rule id="100241" level="13" frequency="20" timeframe="60">
  <if_matched_sid>100240</if_matched_sid>
  <same_field>win.eventdata.clientIPAddress</same_field>
  <description>BloodHound Suspected - Mass LDAP queries from single source</description>
  <mitre><id>T1482</id></mitre>
  <group>attack,discovery</group>
</rule>

<!-- LDAP recon via Sysmon Event 3 (connexions réseau vers port 389/636) -->
<rule id="100242" level="8">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^3$</field>
  <field name="win.eventdata.destinationPort">^(389|636|3268|3269)$</field>
  <field name="win.eventdata.initiated">true</field>
  <description>LDAP Connection from non-DC host - Potential reconnaissance</description>
  <mitre><id>T1018</id></mitre>
</rule>
```

Règles Wazuh — Modifications groupes privilégiés

Toute modification des groupes Domain Admins, Enterprise Admins ou Schema Admins doit générer une alerte de niveau critique immédiate. Ces groupes sont les cibles d'escalade de privilèges les plus directes.

```
 <!-- Ajout membre groupe sécurité global (Domain Admins, etc.) -->
<rule id="100250" level="13">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4728$</field>
  <description>Member Added to Global Security Group - Potential Privilege Escalation</descriptio
  <mitre><id>T1098</id></mitre>
</rule>

<!-- Ajout à groupe local administrateurs sur DC -->
<rule id="100251" level="13">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4732$</field>
  <description>Member Added to Local Administrators Group on DC</description>
  <mitre><id>T1098</id></mitre>
</rule>

<!-- Ajout groupe universel (Enterprise Admins) -->
<rule id="100252" level="14">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^4756$</field>
  <description>Member Added to Universal Security Group - Enterprise Admin escalation risk</descr
  <mitre><id>T1098</id></mitre>
  <group>attack,privilege_escalation</group>
</rule>

<!-- Règle spécifique Domain Admins / Enterprise Admins -->
<rule id="100253" level="15">
  <if_sid>100250,100252</if_sid>
  <list field="win.eventdata.targetUserName" lookup="match_key">etc/lists/privileged-groups</list
  <description>CRITICAL - Member Added to Domain/Enterprise Admins</description>
  <mitre><id>T1098</id></mitre>
  <group>attack,privilege_escalation,critical</group>
</rule>
```

Règles Wazuh — Modifications GPO

Les attaquants qui ont compromis AD modifient souvent les GPO pour déployer des payloads ransomware ou des backdoors sur l'ensemble du parc. L'Event 5136 enregistre les modifications d'attributs AD, dont les objets GPO.

```
 <!-- Modification attribut GPO - Event 5136 -->
<rule id="100260" level="12">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^5136$</field>
  <field name="win.eventdata.objectClass">groupPolicyContainer</field>
  <description>GPO Modified - Potential malicious policy deployment</description>
  <mitre><id>T1484.001</id></mitre>
</rule>

<!-- Création d'une nouvelle GPO -->
<rule id="100261" level="11">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^5137$</field>
  <field name="win.eventdata.objectClass">groupPolicyContainer</field>
  <description>New GPO Created - Verify if authorized change</description>
  <mitre><id>T1484.001</id></mitre>
</rule>

<!-- Liaison GPO à une OU critique -->
<rule id="100262" level="13">
  <if_sid>60007</if_sid>
  <field name="win.system.eventID">^5136$</field>
  <field name="win.eventdata.attributeLDAPDisplayName">gPLink</field>
  <description>GPO Linked to OU - Verify scope of policy application</description>
  <mitre><id>T1484.001</id></mitre>
  <group>attack,defense_evasion</group>
</rule>
```

Intégration Sysmon sur les DC pour enrichir les détections

Sysmon (System Monitor) de Sysinternals enrichit considérablement les capacités de détection sur les DC en loguant les créations de processus, les connexions réseau, et les modifications de clés de registre avec des détails que les événements Windows natifs ne fournissent pas.

```
 <!-- Sysmon Event 1 - Création processus suspect sur DC -->
<rule id="100270" level="12">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^1$</field>
  <field name="win.eventdata.image">(mimikatz|mimilove|wce\.exe|fgdump|pwdump)</field>
  <description>Known Credential Dumper Executed on DC</description>
  <mitre><id>T1003</id></mitre>
</rule>

<!-- Sysmon Event 10 - Accès au processus LSASS -->
<rule id="100271" level="14">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^10$</field>
  <field name="win.eventdata.targetImage">lsass\.exe</field>
  <field name="win.eventdata.grantedAccess">0x1010</field>
  <description>LSASS Memory Access - Credential Dumping Detected</description>
  <mitre><id>T1003.001</id></mitre>
</rule>

<!-- Sysmon Event 11 - Création fichier dans répertoire NTDS -->
<rule id="100272" level="15">
  <if_sid>61600</if_sid>
  <field name="win.system.eventID">^11$</field>
  <field name="win.eventdata.targetFilename">[Nn][Tt][Dd][Ss]</field>
  <description>CRITICAL - File Created in NTDS Directory - Potential NTDS.dit copy</description>
  <mitre><id>T1003.003</id></mitre>
  <group>attack,credential_access,critical</group>
</rule>
```

La configuration Sysmon recommandée pour les DC est disponible sur le projet SwiftOnSecurity/sysmon-config. Pour le déploiement complet, consultez notre guide sur le déploiement de Wazuh SIEM/XDR open source.

Tableau de bord Wazuh recommandé pour AD

Le tableau de bord Wazuh (OpenSearch/Kibana) doit être configuré avec les visualisations suivantes pour une surveillance AD efficace :

Panneau	Métriques	Seuil d'alerte
Tentatives DCSync	Event 4662 avec GUID réplication	1 occurrence
Kerberoasting	Event 4769 etype=0x17 / heure	> 5/heure
NTLM auth. privilégiées	Event 4624 Type3 NTLM comptes Tier0	1 occurrence hors plage horaire
Modifications groupes	Event 4728/4732/4756	Tout changement Domain/ Enterprise Admins
Modifications GPO	Event 5136/5137	Hors fenêtre de maintenance
LDAP Recon	Event 1644 volume/source	> 20 requêtes/min même IP
Accès LSASS	Sysmon Event 10	1 occurrence

Active Response Wazuh — Blocage automatique des attaques AD

La fonctionnalité Active Response de Wazuh permet de déclencher des actions automatiques en réponse à des alertes de sécurité. Pour la protection AD, trois réponses automatiques sont particulièrement utiles : le blocage de compte lors de détection de Kerberoasting massif, l'isolement réseau d'un hôte lors de détection d'accès LSASS, et la notification d'urgence vers le SOC lors d'une tentative DCSync.

Configuration de la réponse automatique — désactivation compte

```
 <!-- ossec.conf sur le manager — active response pour Kerberoasting -->
<ossec_config>
  <command>
    <name>disable-ad-account</name>
    <executable>disable-ad-account.cmd</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>

  <active-response>
    <command>disable-ad-account</command>
    <location>local</location>
    <rules_id>100211</rules_id>
    <timeout>3600</timeout>
  </active-response>
</ossec_config>
```

```
 # disable-ad-account.cmd (script PowerShell appelé par Wazuh Active Response)
# Récupère le nom du compte depuis les données d'alerte Wazuh
param($AlertData)
$accountName = ($AlertData | ConvertFrom-Json).data.win.eventdata.targetUserName
if ($accountName -and $accountName -ne "Administrator") {
    Disable-ADAccount -Identity $accountName
    Write-EventLog -LogName Security -Source "Wazuh-ActiveResponse" `
        -EventId 9999 -EntryType Warning `
        -Message "Account $accountName disabled by Wazuh Active Response (Kerberoasting)"
}
```

Alertes critiques vers le SOC — intégration PagerDuty/Teams

Pour les alertes de niveau 14 et 15 (DCSync, Golden Ticket, accès LSASS), une notification immédiate vers le canal SOC est indispensable. Wazuh propose des intégrations natives avec Slack, Microsoft Teams, PagerDuty et les webhooks génériques.

```
 <!-- Integration Teams pour alertes critiques AD -->
<integration>
  <name>custom-teams</name>
  <hook_url>https://outlook.office.com/webhook/VOTRE-WEBHOOK-TEAMS</hook_url>
  <level>14</level>
  <rule_id>100200,100201,100230,100271,100272</rule_id>
  <alert_format>json</alert_format>
</integration>
```

Tuning et gestion des faux positifs sur les règles AD

Sans tuning approprié, les règles de détection AD génèrent un volume d'alertes qui noie les vrais incidents. Le tuning est un processus itératif qui nécessite au minimum deux semaines de collecte de données avant d'être stabilisé.

Sources courantes de faux positifs AD dans Wazuh

Règle	Source de faux positif	Remédiation
DCSync (100200)	Azure AD Connect, Veeam Backup for AD, ADFS	Whitelist des comptes de service dans la règle
Kerberoasting (100210)	Applications legacy avec RC4 forcé (SQL Server 2008)	Migrer vers AES-256, ou exclure les SPN connus
Pass-the-Hash (100220)	Scripts d'administration réseau avec NTLM	Migrer vers Kerberos, exclure les plages IP d'admin
LDAP Recon (100240)	Outils de gestion AD (RSAT, ADManager Plus)	Exclure les IP des postes d'administration
GPO Modified (100260)	Mises à jour planifiées, déploiements via SCCM	Définir des fenêtres de maintenance, exclure les comptes de déploiement

Exemple de règle de whitelist pour Azure AD Connect

```
 <!-- Exclure Azure AD Connect du DCSync (compte MSOL_xxxxxxxx) -->
<rule id="100203" level="0">
  <if_sid>100200</if_sid>
  <field name="win.eventdata.subjectUserName">^MSOL_</field>
  <description>DCSync from Azure AD Connect - Whitelisted</description>
</rule>

<!-- Exclure les comptes de backup AD -->
<rule id="100204" level="0">
  <if_sid>100200</if_sid>
  <list field="win.eventdata.subjectUserName" lookup="match_key">etc/lists/ad-backup-accounts</li
  <description>DCSync from authorized backup account - Whitelisted</description>
</rule>
```

Métriques de qualité des règles AD

Pour évaluer l'efficacité de votre configuration Wazuh, suivez ces métriques sur une fenêtre glissante de 7 jours : taux de vrais positifs confirmés (objectif supérieur à 95%), temps moyen de détection (objectif inférieur à 5 minutes pour DCSync/Golden Ticket), volume d'alertes niveau 12+ par jour (objectif inférieur à 20 alertes non-tuned après la phase initiale). Un dépassement de ces seuils indique soit un tuning insuffisant, soit une activité malveillante réelle à investiguer. La corrélation avec des données de threat intelligence via MISP, comme décrit dans la section suivante, améliore significativement la précision de la qualification des alertes.

Cas terrain France — SOC PME avec Wazuh + MISP

Intégration MISP pour la corrélation des IoC ANSSI

Une PME industrielle de 800 employés a déployé Wazuh 4.7 sur 6 DC Windows Server 2022 avec une intégration MISP pour la corrélation automatique des Indicateurs de Compromission publiés par l'ANSSI (flux CERT-FR). La configuration utilise le module d'intégration Wazuh-MISP qui enrichit chaque alerte avec les informations de threat intelligence.

```
 # Integration Wazuh-MISP — custom-misp.py (dans /var/ossec/integrations/)
import json, sys, urllib3, requests

urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

MISP_URL = "https://misp.votre-soc.fr"
MISP_KEY = "votre-api-key-misp"

def query_misp(ioc_value):
    headers = {"Authorization": MISP_KEY, "Accept": "application/json", "Content-Type": "applicat
    payload = {"returnFormat": "json", "value": ioc_value, "type": ["ip-src","ip-dst","domain","h
    r = requests.post(f"{MISP_URL}/events/restSearch", headers=headers, json=payload, verify=Fals
    return r.json()

# Lire l'alerte Wazuh depuis stdin
alert = json.loads(sys.stdin.readline())
src_ip = alert.get("data", {}).get("win", {}).get("eventdata", {}).get("ipAddress", "")

if src_ip:
    misp_result = query_misp(src_ip)
    if misp_result.get("response"):
        print(json.dumps({"misp_match": True, "threat_level": misp_result["response"][0]["Event"]
```

Résultats après 6 mois de déploiement

Après 6 mois, le SOC PME a identifié : 3 tentatives de Kerberoasting bloquées automatiquement par active response Wazuh (désactivation du compte source) ; 2 cas de reconnaissance LDAP BloodHound par des pentesteurs autorisés (tests positifs du système de détection) ; 1 alerte Golden Ticket généré lors d'un Red Team interne. Le taux de faux

positifs après tuning a été ramené à moins de 2% pour les règles critiques. La corrélation MISP a permis d'identifier un IP scanneur référencé dans un rapport CERT-FR lié à un groupe APT ciblant l'industrie française. Pour comprendre les frameworks de référence utilisés dans les règles Wazuh, consultez notre article sur MITRE ATT&CK — TTPs 2026.

À RETENIR

Points clés à retenir

Architecture Wazuh pour la surveillance Active Directory

Règles Wazuh pour la détection DCSync

Règles Wazuh pour la détection Kerberoasting

Règles Wazuh pour Pass-the-Hash

Règles Wazuh pour Golden Ticket

FAQ — Détection des Attaques Active Directory avec Wazuh

Wazuh peut-il détecter les attaques Kerberos en temps réel ?

Oui, à condition que l'audit Kerberos soit activé sur les DC (subcategory "Kerberos Service Ticket Operations") et que l'agent Wazuh collecte le canal Security. La détection du Kerberoasting (Event 4769 avec etype RC4) est quasi temps réel, avec un délai typique de 5 à 30 secondes entre l'événement et l'alerte Wazuh. La détection du Golden Ticket est

plus complexe et repose sur des corrélations comportementales plutôt que sur un événement unique.

Quels événements Windows activer sur les DC pour Wazuh ?

Les événements critiques pour la surveillance AD sont : 4662 (accès objet AD avec GUID), 4768/4769 (authentification Kerberos TGT/TGS), 4624/4625 (logon succès/échec), 4728/4732/4756 (modifications groupes), 5136 (modification attribut AD), 1644 (requête LDAP coûteuse). Ces événements nécessitent l'activation de l'audit avancé via auditpol ou GPO. Sans cette configuration, Wazuh ne reçoit pas les événements nécessaires à la détection AD.

Comment différencier un faux positif d'une vraie attaque DCSync ?

Les vraies demandes de réplication DCSync proviennent de comptes qui ne sont pas des contrôleurs de domaine (l'Event 4662 sera généré par un compte utilisateur ou de service, pas par un compte machine DC$). Les faux positifs courants viennent des outils de sauvegarde AD (Veeam, Windows Server Backup) et des solutions de synchronisation d'identité (Azure AD Connect, Okta AD agent). Créez une liste blanche de ces comptes autorisés dans la règle Wazuh via l'opérateur not_match_key pour les exclure de la règle DCSync.

Wazuh supporte-t-il les alertes Sysmon pour les DC ?

Oui, nativement depuis Wazuh 4.2. L'agent Wazuh collecte le canal "Microsoft-Windows-Sysmon/Operational" et des règles pré-configurées (SID 61600-61699) mappent les événements Sysmon vers les techniques MITRE ATT&CK. Déployez Sysmon avec la configuration SwiftOnSecurity sur tous les DC pour enrichir les détections : accès LSASS (Event 10), créations de processus suspects (Event 1), connexions réseau vers les ports LDAP (Event 3), et modifications de registre liées à AD (Event 13).

Quel est le coût de déploiement de Wazuh pour surveiller Active Directory ?

Wazuh est entièrement open source (licence GPLv2). Le coût de déploiement est donc limité à l'infrastructure : un serveur dédié (4 vCPU, 8 Go RAM minimum pour moins de 50 agents), le temps d'intégration (estimé à 3-5 jours pour un environnement de 5 DC avec tuning des règles), et la formation des analystes SOC. Pour les entreprises sans équipe SOC interne, des prestataires MSSP français proposent Wazuh géré à partir de 500-1 500 euros/mois selon le nombre de DC surveillés. Cette solution représente un ROI significatif face aux coûts d'un incident AD non détecté (coût moyen d'un ransomware en France : 500k à 2M euros selon la taille de l'organisation).

Références et documentation

Microsoft Learn — Sécurité Windows Server

ANSSI — Guide de sécurisation Active Directory

MITRE ATT&CK — Techniques Active Directory