Détection des Attaques Active Directory avec Microsoft Sentinel 2026

Guide complet KQL, règles analytiques, MDI et playbooks SOAR pour détecter les attaques Active Directory avec Microsoft Sentinel. Conformité NIS 2.

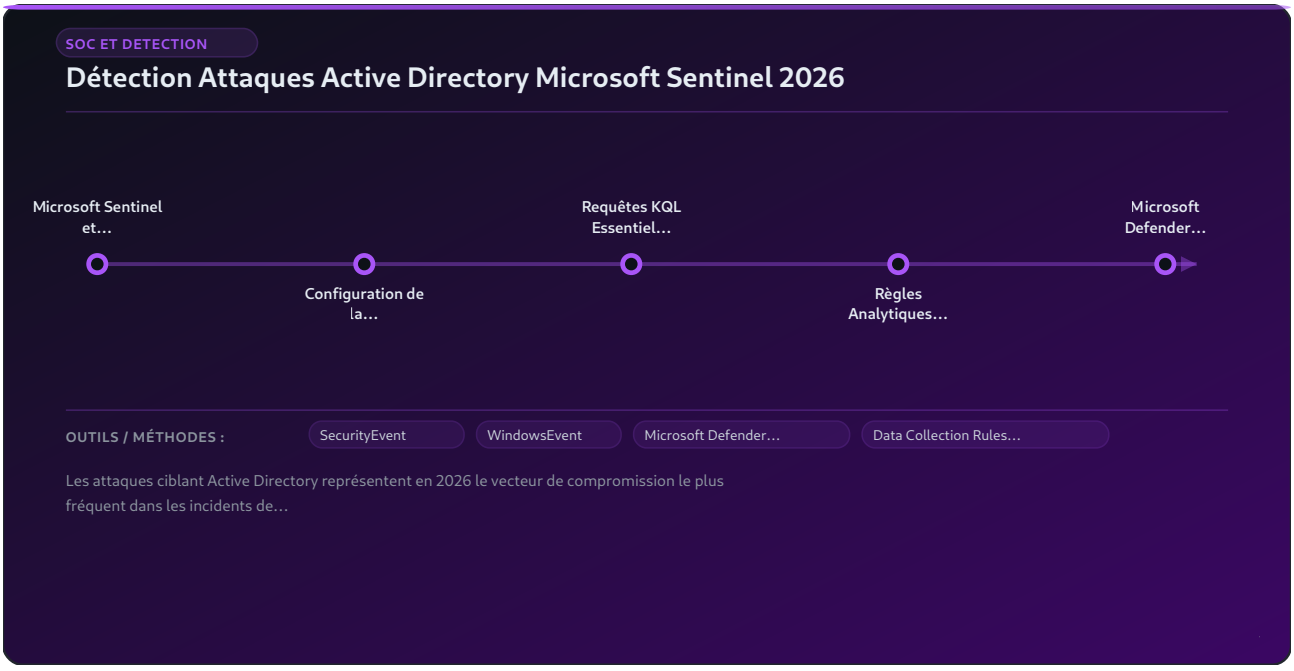
📋 EN BREF

En bref

- Microsoft Sentinel centralise la détection des attaques Active Directory via les tables **SecurityEvent**, **WindowsEvent** et les alertes natives **Microsoft Defender for Identity**.
- 15 requêtes KQL couvrent les attaques AD critiques : DCSync, Kerberoasting, AS-REP Roasting, Pass-the-Hash, Golden Ticket, mouvement latéral et modification AdminSDHolder.
- Les **Data Collection Rules (DCR)** permettent de filtrer les Event IDs à faible valeur et de réduire les coûts d'ingestion de 40 à 60 % sans sacrifier la visibilité.
- Les **Playbooks Logic Apps** automatisent la réponse aux incidents : désactivation de compte compromis, notification Teams et enrichissement via Threat Intelligence.
- La conformité **NIS 2** (Art. 21 et 23) est directement servie par Sentinel via le mapping MITRE ATT&CK, les Incidents horodatés et l'intégration du flux CERT-FR.

Les attaques ciblant Active Directory représentent en 2026 le vecteur de compromission le plus fréquent dans les incidents de cybersécurité traités par les équipes SOC françaises. Selon le

CERT-FR, plus de 70 % des intrusions ayant abouti à un ransomware en France ont impliqué une escalade de privilèges dans Active Directory — DCSync, Golden Ticket ou Kerberoasting — avant le déploiement du chiffreur. Pour les RSSI d'entreprises parisiennes ou de groupes régionaux soumis à la directive NIS 2, la question n'est plus de savoir si Active Directory sera attaqué, mais de combien de temps le SOC disposera pour détecter et contenir l'intrusion avant la compromission totale du domaine. Microsoft Sentinel, le SIEM cloud natif Azure, s'est imposé comme la plateforme de référence pour les organisations qui ont migré vers Microsoft 365 ou Azure, offrant une intégration native avec les sources de logs Windows, Azure Active Directory et Microsoft Defender for Identity. Ce guide technique complet présente l'architecture de détection complète, 15 requêtes KQL opérationnelles, les règles analytiques exportables en ARM template, les playbooks SOAR pour la réponse automatisée, ainsi qu'une analyse des coûts d'ingestion indispensable pour dimensionner le budget SOC dans le contexte des contraintes budgétaires des PME et ETI françaises. Chaque section est ancrée dans le framework MITRE ATT&CK et alignée avec les exigences de la directive NIS 2 transposée en droit français.



Microsoft Sentinel et Active Directory : Architecture de Détection

Microsoft Sentinel est un SIEM (Security Information and Event Management) et SOAR (Security Orchestration, Automation and Response) cloud-native hébergé sur Azure, basé sur Log Analytics. Contrairement aux SIEM on-premise comme Splunk ou QRadar, Sentinel ne nécessite aucune infrastructure à gérer : la scalabilité, la haute disponibilité et les mises à jour sont gérées par Microsoft. Pour la détection AD, l'architecture repose sur trois couches complémentaires.

La première couche est la **collecte des événements Windows** depuis les contrôleurs de domaine. Deux agents coexistent : l'ancien Microsoft Monitoring Agent (MMA, aussi appelé Log Analytics Agent) en fin de vie depuis août 2024, et l'**Azure Monitor Agent (AMA)**, son successeur. AMA s'appuie sur les **Data Collection Rules (DCR)**, des objets Azure ARM qui définissent précisément quels Event IDs collecter, depuis quelles machines, vers quel Log Analytics Workspace. Cette granularité est critique pour maîtriser les coûts : un DC en production peut générer entre 2 000 et 15 000 événements par minute, dont 60 à 70 % sont des événements à faible valeur de sécurité (4634 Logoff, 4648 Explicit credentials).

La deuxième couche est le connecteur **Microsoft Defender for Identity (MDI)**. MDI déploie des capteurs légers directement sur les DC qui analysent le trafic réseau Kerberos, LDAP et DRSUAPI en temps réel. Contrairement aux logs Windows qui nécessitent qu'un auditeur configure les bonnes stratégies d'audit, MDI capture les attaques même quand l'audit Windows est incomplet. Les alertes MDI sont ingérées dans Sentinel via le connecteur natif Microsoft 365 Defender, peuplant la table `SecurityAlert` avec un contexte riche (entités, techniques MITRE, score de confiance).

La troisième couche est **Microsoft Entra ID** (ex-Azure AD) pour les logs d'authentification cloud et la table `IdentityInfo` qui enrichit les requêtes KQL avec les attributs utilisateurs (département, manager, rôles Azure AD), permettant des corrélations entre identités on-premise et cloud.

Les données AD atterrissent dans plusieurs tables Log Analytics : `SecurityEvent` (événements collectés via AMA/MMA), `WindowsEvent` (format XPath plus flexible, recommandé pour les nouvelles DCR), `SecurityAlert` (alertes MDI et Defender), `IdentityDirectoryEvents` (Microsoft 365 Defender Advanced Hunting), `AADSignInLogs` (authentifications Azure AD).

Pour une architecture type avec 3 DC en production, la mise en place nécessite : un Log Analytics Workspace en région France Central (pour la souveraineté des données), l'activation de Sentinel sur ce workspace, l'installation de l'AMA via Azure Arc ou GPO, la création des DCR ciblant les Event IDs critiques, et l'activation du connecteur MDI depuis la page Data Connectors de Sentinel.

Configuration de la Collecte des Events AD dans Sentinel

La configuration du connecteur **Windows Security Events via AMA** est l'étape fondatrice de la détection AD dans Sentinel. Depuis le portail Azure → Microsoft Sentinel → Data Connectors, ce connecteur permet de créer une DCR en quelques clics, mais la configuration par défaut "All Security Events" est à éviter en production : elle collecte les 300+ Event IDs de sécurité Windows, générant un volume et un coût inutiles.

> ⚡ **Retour terrain**
>
> Pour un groupe de services financiers qui recevait 2 millions d'alertes SIEM par jour et en traitait 3 %, j'ai réalisé une analyse de la valeur des règles de détection sur 6 mois de logs. Résultat : 73 % des alertes provenaient de 4 règles mal calibrées. La suppression ou le réajustement de ces 4 règles a réduit le bruit de 65 % sans manquer aucun incident réel lors des 3 mois de validation. Le ROI d'un tuning SIEM est parmi les meilleurs investissements défensifs possibles.

La configuration recommandée pour un SOC AD passe par une **DCR personnalisée** avec les filtres XPath suivants, appliqués au canal `Security` :

```
 <!-- DCR - Security Events critiques AD -->
<WindowsEventLog>
  <CollectorType>Security</CollectorType>
  <XPathQueries>
    <XPathQuery>Security!*[System[(EventID=4624 or EventID=4625 or EventID=4648)]]</XPathQuery>
    <XPathQuery>Security!*[System[(EventID=4662 or EventID=4663)]]</XPathQuery>
    <XPathQuery>Security!*[System[(EventID=4720 or EventID=4722 or EventID=4725 or EventID=4726 o
    <XPathQuery>Security!*[System[(EventID=4728 or EventID=4729 or EventID=4732 or EventID=4756 o
    <XPathQuery>Security!*[System[(EventID=4768 or EventID=4769 or EventID=4771 or EventID=4776)]
    <XPathQuery>Security!*[System[(EventID=5136 or EventID=5141)]]</XPathQuery>
    <XPathQuery>System!*[System[(EventID=7045 or EventID=7036)]]</XPathQuery>
  </XPathQueries>
</WindowsEventLog>
```

Le tableau ci-dessous synthétise les Event IDs indispensables, leur table Sentinel cible, l'attaque détectée et la priorité de collecte :

Event ID	Table Sentinel	Attaque ciblée	MITRE ATT&CK	Priorité
4662	SecurityEvent	DCSync (réplication AD)	T1003.006	Critique
4769	SecurityEvent	Kerberoasting (TGS RC4)	T1558.003	Critique
4768	SecurityEvent	AS-REP Roasting / Golden Ticket	T1558.001/4	Critique
4624 (Type 3)	SecurityEvent	Pass-the-Hash, mouvement latéral	T1550.002	Haute
4625	SecurityEvent	Password Spray, Brute Force	T1110.003	Haute
4728 / 4729	SecurityEvent	Ajout compte Domain Admins	T1098	Haute
4663	SecurityEvent	Accès NTDS.dit (dump credentials)	T1003.003	Critique
5136	SecurityEvent	Modification AdminSDHolder	T1484.001	Haute
7045	WindowsEvent (System)	Installation service (PsExec)	T1021.002	Haute
4771	SecurityEvent	Pre-auth Kerberos échouée	T1558.004	Moyenne
4776	SecurityEvent	NTLM auth (Pass-the-Hash)	T1550.002	Haute
5141	SecurityEvent	Suppression objet AD (couverture)	T1070	Moyenne

Pour le connecteur **Microsoft Defender for Identity**, l'activation se fait depuis Sentinel → Data Connectors → Microsoft 365 Defender. Une fois connecté, les alertes MDI apparaissent dans `SecurityAlert` avec `ProductName = "Azure Advanced Threat Protection"` (nom legacy conservé dans l'API). La table `IdentityDirectoryEvents` (disponible via le connecteur M365D Advanced Hunting) est particulièrement utile pour les requêtes sur les modifications d'objets AD sans

passer par les Event IDs Windows. Consultez également notre article sur les événements Windows essentiels à surveiller sur les contrôleurs de domaine pour une référence complète des Event IDs.

Requêtes KQL Essentielles pour la Détection AD

Le KQL (Kusto Query Language) est le langage de requête de Sentinel. Les 15 requêtes suivantes couvrent les principales techniques d'attaque AD du framework MITRE ATT&CK. Chaque requête est accompagnée d'un commentaire explicatif et peut être directement utilisée comme base pour une Analytic Rule.

```
// ============================================================================
// 1. DCSYNC ATTACK DETECTION (MITRE T1003.006)
// Détecte les demandes de réplication AD depuis un compte non-machine
// Event 4662: opération sur objet AD avec droits de réplication
// ============================================================================
SecurityEvent
| where EventID == 4662
| where ObjectType contains "19195a5b-6da0-11d0-afd3-00c04fd930c9"  // domainDNS
| where Properties has "1131f6aa-9c07-11d1-f79f-00c04fc2dcd2"        // DS-Replication-Get-Change
    or Properties has "1131f6ab-9c07-11d1-f79f-00c04fc2dcd2"        // DS-Replication-Get-Changes
| where SubjectUserName !endswith "$"  // Exclure comptes machines (réplication légitime)
| project TimeGenerated, SubjectUserName, SubjectDomainName,
          IpAddress, Computer, Properties
| extend AlertSeverity = "High", ATT_CK = "T1003.006"
```

```
// ============================================================================
// 2. KERBEROASTING DETECTION (MITRE T1558.003)
// Détecte les demandes TGS avec chiffrement RC4 (0x17) pour des comptes
// de service - signature classique de Kerberoasting
// ============================================================================
SecurityEvent
| where EventID == 4769
| where TicketEncryptionType == "0x17"  // RC4-HMAC - chiffrement faible
| where ServiceName !endswith "$"       // Exclure comptes machines
| where ServiceName != "krbtgt"         // Exclure le compte krbtgt lui-même
| where ClientAddress !in ("::1", "127.0.0.1")  // Exclure loopback
| summarize
    RequestCount = count(),
    TargetAccounts = make_set(ServiceName),
    SourceIPs = make_set(ClientAddress)
    by bin(TimeGenerated, 1h), TargetUserName, ClientAddress
| where RequestCount > 3  // Seuil: plus de 3 TGS RC4 en 1h = suspect
| extend AlertSeverity = "High", ATT_CK = "T1558.003"
```

```
// ============================================================================
// 3. AS-REP ROASTING DETECTION (MITRE T1558.004)
// Détecte les comptes avec PreAuth désactivée (0x410200 dans UserAccountControl)
// et les TGT demandés sans pre-authentification
// ============================================================================
SecurityEvent
| where EventID == 4768
| where PreAuthType == "0"         // Pre-authentication non requise
| where ResultCode == "0x0"        // Succès (le hash AS-REP est retourné)
| where TargetUserName != "ANONYMOUS LOGON"
| project TimeGenerated, TargetUserName, TargetDomainName,
          IpAddress, Computer
| extend AlertSeverity = "High", ATT_CK = "T1558.004",
    Description = strcat("Compte sans pré-authentification Kerberos : ", TargetUserName)
```

```
// ========================================================================
// 4. PASS-THE-HASH DETECTION (MITRE T1550.002)
// LogonType=3 (réseau) avec NTLM et WorkstationName vide = PtH suspect
// Corrélation: 4624 (succès) et 4776 (NTLM validation)
// ========================================================================
SecurityEvent
| where EventID == 4624
| where LogonType == 3                    // Logon réseau
| where AuthenticationPackageName == "NTLM"
| where LogonProcessName == "NtLmSsp"
| where WorkstationName == ""             // Typique des outils PtH (Mimikatz)
| where TargetUserName !endswith "$"
| where TargetUserName != "ANONYMOUS LOGON"
| project TimeGenerated, TargetUserName, TargetDomainName,
          IpAddress, WorkstationName, Computer, LogonGuid
| extend AlertSeverity = "High", ATT_CK = "T1550.002"
```

```
// ========================================================================
// 5. PASSWORD SPRAY DETECTION (MITRE T1110.003)
// Détecte un attaquant qui essaie le même mot de passe sur de nombreux comptes
// différents depuis la même IP - faible nombre d'échecs par compte
// ========================================================================
SecurityEvent
| where EventID == 4625          // Authentication failure
| where LogonType in (2, 3)
| where AccountType == "User"
| summarize
    FailedAccounts = dcount(TargetUserName),
    AttemptCount = count(),
    AccountsList = make_set(TargetUserName, 10)
    by bin(TimeGenerated, 30m), IpAddress
| where FailedAccounts >= 10     // 10+ comptes distincts depuis la même IP
| where AttemptCount / FailedAccounts < 5  // Peu d'essais par compte = spray
| project TimeGenerated, IpAddress, FailedAccounts, AttemptCount, AccountsList
| extend AlertSeverity = "Medium", ATT_CK = "T1110.003"
```

```
// ========================================================================
// 6. GOLDEN TICKET DETECTION (MITRE T1558.001)
// Les Golden Tickets créés par Mimikatz ont souvent une durée de vie
// anormalement longue (>10h) ou des TicketOptions non-standard
// ========================================================================
SecurityEvent
| where EventID == 4769
| where ServiceName == "krbtgt"
| extend TicketLifetimeHours = datetime_diff('hour',
    todatetime(TicketOptions), TimeGenerated)
// Détecter les TGS pour krbtgt depuis des IPs inattendues
| where ClientAddress !in
    (toscalar(
        SecurityEvent
        | where EventID == 4768
        | where TimeGenerated > ago(30d)
        | summarize make_set(IpAddress) by Computer
        | summarize make_set_if(set_IpAddress, Computer !endswith "DC")
    ))
| project TimeGenerated, TargetUserName, ClientAddress,
          TicketEncryptionType, TicketOptions, Computer
| extend AlertSeverity = "Critical", ATT_CK = "T1558.001"
```

```
// ========================================================================
// 7. DOMAIN ADMIN GROUP MEMBERSHIP CHANGE (MITRE T1098)
// Alerte immédiate dès qu'un compte est ajouté aux Domain Admins
// ou tout autre groupe privilégié (Enterprise Admins, Schema Admins)
// ========================================================================
SecurityEvent
| where EventID in (4728, 4732, 4756)  // Ajout membre groupe (global/local/universel)
| where MemberName != "-"
| where TargetUserName in ("Domain Admins", "Enterprise Admins",
                           "Schema Admins", "Group Policy Creator Owners",
                           "Administrateurs du schéma", "Administrateurs de l'entreprise")
| project TimeGenerated, SubjectUserName, SubjectDomainName,
          MemberName, TargetUserName, Computer
| extend AlertSeverity = "Critical", ATT_CK = "T1098",
    Description = strcat(SubjectUserName, " a ajouté ", MemberName,
                         " dans le groupe ", TargetUserName)
```

```
// ========================================================================
// 8. NTDS.DIT ACCESS DETECTION (MITRE T1003.003)
// Accès direct au fichier NTDS.dit = tentative de dump de la base AD
// Nécessite l'audit d'accès aux objets sur SYSVOL/NTDS activé
// ========================================================================
SecurityEvent
| where EventID == 4663
| where ObjectName has "ntds.dit" or ObjectName has "NTDS.DIT"
| where ProcessName !has "ntdsutil"  // Exclure sauvegarde légitime (à adapter)
| project TimeGenerated, SubjectUserName, SubjectDomainName,
          ObjectName, ProcessName, IpAddress, Computer
| extend AlertSeverity = "Critical", ATT_CK = "T1003.003",
    Description = "Accès direct au fichier NTDS.dit détecté"
```

```
// ========================================================================
// 9. ADMINSDH OLDER MODIFICATION (MITRE T1484.001)
// L'AdminSDHolder est un objet AD dont les ACL sont propagées à tous
// les comptes privilégiés - sa modification est un vecteur de persistance
// ========================================================================
SecurityEvent
| where EventID == 5136  // A directory service object was modified
| where ObjectDN has "CN=AdminSDHolder"
| project TimeGenerated, SubjectUserName, SubjectDomainName,
          ObjectDN, AttributeLDAPDisplayName, AttributeValue, Computer
| extend AlertSeverity = "Critical", ATT_CK = "T1484.001",
    Description = strcat("Modification AdminSDHolder par ", SubjectUserName,
                         " - attribut: ", AttributeLDAPDisplayName)
```

```
// ========================================================================
// 10. LATERAL MOVEMENT VIA PSEXEC/REMOTE SERVICES (MITRE T1021.002)
// PsExec installe un service temporaire (PSEXESVC) détectable via 7045
// Corrélation avec 4624 LogonType=3 depuis la même source
// ========================================================================
let ServiceInstalls = SecurityEvent
| where EventID == 7045
| where ServiceName has "PSEXESVC" or ServiceFileName has "psexec"
       or ServiceFileName has "\\127.0.0.1\"  // Service via partage admin
| project InstallTime = TimeGenerated, ServiceComputer = Computer,
          ServiceName, ServiceFileName, SubjectUserName;
let NetworkLogons = SecurityEvent
| where EventID == 4624
| where LogonType == 3
| project LogonTime = TimeGenerated, LogonComputer = Computer,
          TargetUserName, IpAddress;
ServiceInstalls
| join kind=inner (NetworkLogons) on $left.ServiceComputer == $right.LogonComputer
| where abs(datetime_diff('minute', InstallTime, LogonTime)) < 5
| project InstallTime, ServiceComputer, ServiceName, SubjectUserName, IpAddress
| extend AlertSeverity = "High", ATT_CK = "T1021.002"
```

```
// ========================================================================
// 11. SUSPICIOUS SCHEDULED TASK CREATION (MITRE T1053.005)
// Tâches planifiées créées par des comptes non-système en dehors des
// heures ouvrées ou sur des DC (inhabituel)
// ========================================================================
SecurityEvent
| where EventID == 4698  // A scheduled task was created
| where SubjectUserName !endswith "$"
| where SubjectUserName !in ("SYSTEM", "LOCAL SERVICE", "NETWORK SERVICE")
| extend TaskHour = hourofday(TimeGenerated)
| where TaskHour !between (8 .. 18)  // Hors heures ouvrées
    or Computer endswith "DC01" or Computer endswith "DC02"  // Sur DC = suspect
| project TimeGenerated, SubjectUserName, SubjectDomainName,
          Computer, TaskName, TaskContent
| extend AlertSeverity = "Medium", ATT_CK = "T1053.005"
```

```
// ============================================================================
// 12. HONEY ACCOUNT ACCESS (DECEPTION - T1078)
// Jointure avec la Watchlist Sentinel "HoneyAccounts"
// Tout accès à ces comptes appâts déclenche une alerte critique
// ============================================================================
let HoneyAccounts = _GetWatchlist('HoneyAccounts')
    | project SearchKey, AccountName, Description;
SecurityEvent
| where EventID in (4624, 4625, 4768, 4769)
| join kind=inner (HoneyAccounts) on $left.TargetUserName == $right.AccountName
| project TimeGenerated, EventID, TargetUserName, IpAddress,
          Computer, LogonType, Description
| extend AlertSeverity = "Critical", ATT_CK = "T1078",
    Description = strcat("Accès au honey account : ", TargetUserName,
                         " depuis ", IpAddress)
```

```
// ============================================================================
// 13. ANOMALOUS PRIVILEGED ACCOUNT ACTIVITY (T1078.002)
// Détecte les comptes Domain Admins qui se connectent depuis des postes
// non-habituels (référence: baseline 30 derniers jours)
// ============================================================================
let PrivilegedAccounts = _GetWatchlist('PrivilegedAccounts')
    | project AccountName;
let Baseline = SecurityEvent
| where TimeGenerated between (ago(30d) .. ago(1d))
| where EventID == 4624
| join kind=inner (PrivilegedAccounts) on $left.TargetUserName == $right.AccountName
| summarize BaselineComputers = make_set(WorkstationName) by TargetUserName;
SecurityEvent
| where TimeGenerated > ago(1d)
| where EventID == 4624
| join kind=inner (PrivilegedAccounts) on $left.TargetUserName == $right.AccountName
| join kind=leftouter (Baseline) on TargetUserName
| where WorkstationName !in (BaselineComputers)  // Nouveau poste
| project TimeGenerated, TargetUserName, WorkstationName, IpAddress, BaselineComputers
| extend AlertSeverity = "High", ATT_CK = "T1078.002"
```

```
// ========================================================================
// 14. ACCOUNT CREATION OUTSIDE BUSINESS HOURS (T1136.001)
// Création de comptes AD en dehors des plages autorisées -
// indicateur d'activité malveillante après compromission initiale
// ========================================================================
SecurityEvent
| where EventID == 4720  // A user account was created
| where SubjectUserName !endswith "$"
| extend CreateHour = hourofday(TimeGenerated),
         CreateDay = dayofweek(TimeGenerated)
| where CreateHour !between (8 .. 18)              // Hors heures ouvrées
    or CreateDay in (6, 0)                          // Week-end (Sam=6, Dim=0)
| project TimeGenerated, SubjectUserName, SubjectDomainName,
          TargetUserName, Computer
| extend AlertSeverity = "Medium", ATT_CK = "T1136.001",
    Description = strcat("Compte créé hors heures ouvrées: ", TargetUserName,
                         " par ", SubjectUserName)
```

```
// ========================================================================
// 15. RECON VIA LDAP ENUMERATION (T1087.002)
// Volume anormalement élevé de requêtes LDAP depuis un seul hôte
// MDI détecte cela nativement, mais cette KQL complémente via DNS/ETW
// ========================================================================
// Via SecurityAlert (alertes MDI natives)
SecurityAlert
| where ProductName == "Azure Advanced Threat Protection"
| where AlertName has_any ("Reconnaissance", "LDAP", "Enumeration",
                           "Account enumeration", "Network mapping")
| project TimeGenerated, AlertName, Severity, Entities,
          ProviderAlertId, RemediationSteps
| extend ATT_CK = "T1087.002"
| order by TimeGenerated desc
```

Ces requêtes s'appuient sur les principes décrits dans notre article sur la [détection des attaques Active Directory avec Wazuh], qui couvre la même problématique pour les environnements open-source.

Règles Analytiques Sentinel pour Active Directory

Les requêtes KQL deviennent des **Analytic Rules** dans Sentinel : elles s'exécutent périodiquement, créent des Incidents et peuvent déclencher des Playbooks. La configuration d'une règle analytique comporte plusieurs paramètres critiques :

Rule frequency : toutes les 5 minutes pour les attaques critiques (DCSync, Golden Ticket), toutes les heures pour Password Spray

Lookback period : fenêtre d'analyse, généralement 1h à 24h selon la technique

Alert threshold : nombre minimum d'événements pour générer une alerte (éviter les faux positifs)

Entity mapping : mapper les champs KQL vers les entités Sentinel (Account, IP, Host) pour enrichir l'Incident

MITRE ATT&CK tactics/techniques : tagging pour la vue MITRE dans Sentinel

Voici un exemple d'**ARM template JSON exportable** pour la règle de détection DCSync :

```
{
 "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {
   "workspace": {
     "type": "string",
     "metadata": { "description": "Nom du Log Analytics Workspace Sentinel" }
   }
 },
 "resources": [
   {
     "type": "Microsoft.OperationalInsights/workspaces/providers/alertRules",
     "apiVersion": "2022-11-01-preview",
     "name": "[concat(parameters('workspace'), '/Microsoft.SecurityInsights/DCSync-Detection-Rul
     "kind": "Scheduled",
     "properties": {
       "displayName": "[CRITICAL] DCSync Attack Detected",
       "description": "Détecte les demandes de réplication DCSync depuis un compte non-machine.
       "severity": "High",
       "enabled": true,
       "query": "SecurityEvent | where EventID == 4662 | where ObjectType contains '19195a5b-6da
       "queryFrequency": "PT5M",
       "queryPeriod": "PT1H",
       "triggerOperator": "GreaterThan",
       "triggerThreshold": 0,
       "suppressionDuration": "PT1H",
       "suppressionEnabled": false,
       "tactics": ["CredentialAccess"],
       "techniques": ["T1003"],
       "subTechniques": ["T1003.006"],
       "entityMappings": [
         {
           "entityType": "Account",
           "fieldMappings": [
             { "identifier": "Name", "columnName": "SubjectUserName" },
             { "identifier": "NTDomain", "columnName": "SubjectDomainName" }
           ]
         },
         {
           "entityType": "IP",
           "fieldMappings": [
             { "identifier": "Address", "columnName": "IpAddress" }
           ]
```

```
          },
          {
            "entityType": "Host",
            "fieldMappings": [
              { "identifier": "HostName", "columnName": "Computer" }
            ]
          }
        ],
        "alertDetailsOverride": {
          "alertDisplayNameFormat": "DCSync depuis {{SubjectUserName}} ({{IpAddress}})",
          "alertDescriptionFormat": "Le compte {{SubjectUserName}} a effectué une demande de répl
        },
        "incidentConfiguration": {
          "createIncident": true,
          "groupingConfiguration": {
            "enabled": true,
            "reopenClosedIncident": false,
            "lookbackDuration": "PT5H",
            "matchingMethod": "Selected",
            "groupByEntities": ["Account", "IP"]
          }
        }
      }
    }
  ]
}
```

Ce template peut être déployé via Azure CLI (`az deployment group create`), Azure DevOps ou GitHub Actions dans une pipeline Detection-as-Code. Consultez notre guide sur [la configuration Sysmon sur les contrôleurs de domaine] pour enrichir davantage la télémétrie disponible dans Sentinel.

Microsoft Defender for Identity (MDI) : Alertes Natives

Microsoft Defender for Identity complète Sentinel avec une détection comportementale spécialisée sur Active Directory. Les capteurs MDI, installés directement sur les DC, capturent le trafic Kerberos, LDAP, RPC/DRSUAPI et DNS en temps réel, sans dépendre de la politique

d'audit Windows. Cette architecture permet à MDI de détecter des attaques qui laissent peu ou pas de traces dans les Event Logs.

Les **alertes MDI clés pour les équipes SOC** en 2026 couvrent :

Suspected DCSync attack : détection DRSUAPI depuis un hôte non-DC

Suspected Golden Ticket usage : anomalie dans les attributs du ticket Kerberos (durée, chiffrement, PAC)

Suspected Kerberoasting : pic de demandes TGS RC4 sur comptes de service

Account enumeration reconnaissance : LDAP queries volumineuses depuis un seul hôte

Suspected Overpass-the-Hash : utilisation d'un hash NTLM pour demander un ticket Kerberos

Suspected skeleton key attack : modification en mémoire du processus lsass.exe du DC

Suspected DFSCoerce / PetitPotam : coercion d'authentification NTLM vers un relais

Lateral movement path : visualisation graphique des chemins de compromission possibles

Dans Sentinel, les alertes MDI sont interrogeables via la KQL suivante pour une vue unifiée :

```
// Toutes les alertes MDI des dernières 24h, triées par sévérité
SecurityAlert
| where TimeGenerated > ago(24h)
| where ProductName == "Azure Advanced Threat Protection"
| extend Entities_parsed = parse_json(Entities)
| project TimeGenerated, AlertName, Severity,
          AlertSeverity, Description, RemediationSteps,
          Entities_parsed, ProviderAlertId
| order by case(
    AlertSeverity == "High", 1,
    AlertSeverity == "Medium", 2,
    3) asc, TimeGenerated desc
```

La table `IdentityDirectoryEvents` (Microsoft 365 Defender Advanced Hunting, disponible dans Sentinel via le connecteur M365D) permet des requêtes sur les actions AD comme les changements de mot de passe, les modifications de groupes et les créations de comptes, avec un niveau de détail supérieur aux Event Logs Windows :

```
// Modifications de groupe via Advanced Hunting MDI
IdentityDirectoryEvents
| where TimeGenerated > ago(7d)
| where ActionType in ("Group Membership changed", "User account created",
                       "User account deleted", "Password changed")
| where DestinationDeviceName != ""
| project TimeGenerated, ActionType, AccountUpn, AccountDisplayName,
          AdditionalFields, DestinationDeviceName
| order by TimeGenerated desc
```

Pour les organisations déjà équipées de honeypots AD, notre article sur les honey accounts et SPNs Active Directory détaille comment créer des comptes appâts dont les accès sont détectés par cette infrastructure. Voir aussi notre guide sur les Authentication Silos Windows Server 2025 pour segmenter les comptes Tier 0 et réduire la surface d'attaque détectable.

Playbooks SOAR : Réponse Automatisée aux Incidents AD

Les Playbooks Sentinel sont des **Azure Logic Apps** déclenchés automatiquement lorsqu'un Incident est créé ou mis à jour. Pour les attaques AD, trois playbooks types couvrent 80 % des besoins de réponse automatisée.

Playbook 1 — Isolation de compte compromis (Disable-ADAccount via Azure Automation) :

```
{
  "definition": {
    "$schema": "https://schema.management.azure.com/providers/Microsoft.Logic/schemas/2016-06-01/
    "triggers": {
      "Microsoft_Sentinel_incident": {
        "type": "ApiConnectionWebhook",
        "inputs": {
          "body": { "callback_url": "@{listCallbackUrl()}" },
          "host": {
            "connection": { "name": "@parameters('$connections')['azuresentinel']['connectionId']
          },
          "path": "/incident-creation"
        }
      }
    },
    "actions": {
      "Get_Incident_Accounts": {
        "type": "ApiConnection",
        "inputs": {
          "host": {
            "connection": { "name": "@parameters('$connections')['azuresentinel']['connectionId']
          },
          "method": "post",
          "path": "/Incidents/subscriptions/@{encodeURIComponent(triggerBody()?['WorkspaceSubscri
          "body": { "LookbackInMinutes": 60 }
        }
      },
      "For_Each_Account": {
        "type": "Foreach",
        "foreach": "@body('Get_Incident_Accounts')?['Entities']",
        "actions": {
          "Disable_AD_Account_via_Automation": {
            "type": "ApiConnection",
            "inputs": {
              "host": {
                "connection": { "name": "@parameters('$connections')['azureautomation']['connecti
              },
              "method": "put",
              "path": "/subscriptions/@{encodeURIComponent('SUBSCRIPTION_ID')}/resourceGroups/@{e
              "body": {
                "properties": {
                  "runbook": { "name": "Disable-CompromisedADAccount" },
                  "parameters": {
```

```
                    "UserName": "@{items('For_Each_Account')?['Name']}",
                    "Domain": "@{items('For_Each_Account')?['NTDomain']}"
                  }
                }
              }
            }
          },
          "Notify_Teams": {
            "type": "ApiConnection",
            "inputs": {
              "host": {
                "connection": { "name": "@parameters('$connections')['teams']['connectionId']" }
              },
              "method": "post",
              "path": "/v3/beta/teams/@{encodeURIComponent('TEAM_ID')}/channels/@{encodeURICompon
              "body": {
                "body": {
                  "content": "🚨 **COMPTE AD DÉSACTIVÉ** — Incident Sentinel : @{triggerBody()?['

Compte : **@{items('For_Each_Account')?['Name']}@{items('For_Each_Account')?['NTDomain']}**

Raison : @{triggerBody()?['object']?['properties']?['title']}

Action : Compte désactivé automatiquement. Vérification manuelle requise avant réactivation."
                }
              }
            }
          }
        }
      },
      "Add_Comment_to_Incident": {
        "type": "ApiConnection",
        "inputs": {
          "host": {
            "connection": { "name": "@parameters('$connections')['azuresentinel']['connectionId']
          },
          "method": "post",
          "path": "/Incidents/subscriptions/@{encodeURIComponent(triggerBody()?['WorkspaceSubscri
          "body": {
            "message": "Playbook exécuté automatiquement : comptes compromis désactivés dans Acti
          }
        }
      }
    }
```

```
  }
}
```

Playbook 2 — Enrichissement IP via Threat Intelligence : Ce playbook interroge automatiquement les APIs VirusTotal, AbuseIPDB et le flux TAXII du CERT-FR pour enrichir l'IP source d'une alerte. Le résultat est ajouté en commentaire de l'Incident, permettant à l'analyste de prioriser sa réponse sans quitter Sentinel.

Playbook 3 — Escalade automatique NIS 2 : Déclenché sur les Incidents de sévérité High/Critical impliquant des comptes Tier 0 ou des données classifiées, ce playbook génère automatiquement un brouillon de notification d'incident au format ANSSI (formulaire CERRF-IN-001), pré-rempli avec la timeline et les entités de l'Incident Sentinel, et l'envoie par email à l'équipe RSSI pour validation avant transmission.

Watchlists Sentinel : Listes de Comptes et Assets Sensibles

Les **Watchlists** Sentinel sont des tables de référence importées en CSV ou JSON, interrogeables dans n'importe quelle KQL via la fonction `_GetWatchlist('NomWatchlist')`. Pour la détection AD, trois watchlists sont indispensables.

Watchlist "PrivilegedAccounts" — liste des comptes Tier 0 et Tier 1 :

```
SearchKey,AccountName,Tier,Description,Owner
admin-dc01,admin-dc01,0,Compte local DC01,équipe infra
svc-backup,svc-backup,1,Compte service backup Veeam,équipe backup
krbtgt,krbtgt,0,Compte Kerberos TGT,AD
CORP\Administrateur,Administrateur,0,Compte admin domaine par défaut,AD
```

Watchlist "HoneyAccounts" — comptes appâts qui ne devraient jamais être utilisés :

```
SearchKey,AccountName,Description,CreatedDate
svc-legacy-sql,svc-legacy-sql,Honey SPN - ancienne app SQL fictive,2026-01-15
admin-old-dc,admin-old-dc,Honey admin - DC fictif désaffecté,2026-01-15
backup-svc-2019,backup-svc-2019,Honey service backup,2026-01-15
```

La jointure dans les requêtes KQL (comme illustré dans la requête #12 ci-dessus) enrichit automatiquement les alertes avec le contexte de la watchlist : tier de l'actif, propriétaire, description. Cette approche est décrite plus en détail dans notre article sur les honeypots Active Directory.

Watchlist "TrustedAdminWorkstations" — postes autorisés pour les connexions admin (PAW) : tout accès admin depuis un poste absent de cette liste déclenche une alerte immédiate, couvrant la technique T1078.002 (Valid Accounts: Domain Accounts) en cas de compromission des identifiants d'un administrateur.

Dashboard et Workbooks Sentinel pour Active Directory

Microsoft Sentinel propose la galerie **Workbooks** avec des tableaux de bord préconfigurés. Pour Active Directory, le workbook **"Azure AD Audit, Activity and Sign-in logs"** et le workbook communautaire **"Active Directory Security"** (disponible sur GitHub microsoft/Azure-Sentinel) couvrent les métriques essentielles.

Les métriques SOC clés à monitorer en temps réel pour AD :

MTTD (Mean Time to Detect) : temps moyen entre l'Event ID 4662 (DCSync) et la création de l'Incident Sentinel. Objectif : < 5 minutes.

Taux d'échecs Kerberos (4768/4769 ResultCode != 0x0) : baseline sur 30j, alerte si +3 sigma en 1h

Comptes avec échecs auth (>10/heure) : top 10 des comptes les plus ciblés

Modifications groupes privilégiés (24h) : compteur zéro-based (tout changement est notable)

Volume d'ingestion DC par jour (Go) : tendance hebdo pour anticiper les coûts

Incidents ouverts par technique MITRE : heatmap ATT&CK pour identifier les lacunes de couverture

Pour créer un workbook personnalisé centré sur AD, voici une requête KQL pour la visualisation des tendances d'authentification :

```
// Tendance des authentifications Kerberos par heure (succès vs échecs)
SecurityEvent
| where EventID in (4768, 4769, 4771)
| where TimeGenerated > ago(7d)
| summarize
    Successes = countif(ResultCode == "0x0"),
    Failures = countif(ResultCode != "0x0")
    by bin(TimeGenerated, 1h), EventID
| extend EventName = case(
    EventID == 4768, "TGT Request",
    EventID == 4769, "TGS Request",
    "Pre-auth Failure")
| render timechart with (title="Activité Kerberos - 7 derniers jours")
```

Coûts et Optimisation : Filtrer les Events sans Perdre la Visibilité

Le coût d'ingestion Sentinel est le principal frein à son adoption dans les PME françaises. Voici une estimation réaliste pour une infrastructure AD typique :

Scénario	Go/DC/jour	3 DC/mois	Coût mensuel estimé*
Tous les Security Events (défaut)	3–5 Go	270–450 Go	660–1 100 €
DCR avec Event IDs critiques uniquement	0,8–1,5 Go	72–135 Go	175–330 €
DCR + Transformation Rules (enrichissement)	0,6–1,2 Go	54–108 Go	130–265 €
Commitment Tier 100 Go/jour (annuel)	—	—	~125 € (flat)

**Tarifs indicatifs Europe Ouest, Pay-as-you-go ~2,46 €/Go. Hors coûts MDI (~5,20 $/user/mois dans M365 Defender P2).*

Les événements à **exclure absolument** des DCR pour réduire le volume sans perte de visibilité sécurité :

4634 (An account was logged off) : aucune valeur detective, volume très élevé

4648 (A logon was attempted using explicit credentials) : génère des faux positifs massifs sur les services Windows normaux

5145 (A network share object was checked) : bruit réseau sans signification sécurité seul

4703 (A token right was adjusted) : volume très élevé sur les DCs actifs

4907 (Auditing settings on object were changed) : modifiable légitimement, faible valeur seul

Les **Transformation Rules** (DCR Transformations, en GA depuis fin 2024) permettent d'exécuter une requête KQL au moment de l'ingestion pour enrichir ou filtrer les événements avant qu'ils ne soient stockés dans Log Analytics, réduisant le volume facturé et améliorant la qualité des données.

Conformité NIS 2 : Sentinel comme SOC-as-a-Service pour Entreprises Françaises

La directive NIS 2, transposée en France par la loi n°2024-449 du 21 mai 2024, impose aux entités essentielles et importantes des obligations de sécurité et de notification renforcées. Microsoft Sentinel répond directement à deux articles clés.

Article 21 — Mesures de gestion des risques cyber : les Analytic Rules Sentinel mappées sur MITRE ATT&CK constituent une preuve documentée de la capacité de détection. Le Content Hub Sentinel propose une solution "NIS2 Compliance" qui installe automatiquement les règles couvrant les exigences de l'article 21, notamment la détection des accès non autorisés, la surveillance des comptes privilégiés et la détection des malwares.

Article 23 — Notification des incidents : toute entité doit notifier l'ANSSI sous 24h après détection d'un incident significatif. Sentinel facilite cette obligation via :

Les Incidents avec **timeline complète** (premier événement, détection, actions de réponse) exportable en JSON pour le rapport ANSSI

Le connecteur **TAXII CERT-FR** (feed MISP disponible sur `https://misp.cert.ssi.gouv.fr/`) qui ingère les IoC publiés par le CERT-FR dans la table `ThreatIntelligenceIndicator`, permettant une corrélation automatique avec les événements AD

Les **Workbooks de conformité** qui génèrent des rapports auditables avec preuves d'activité de monitoring

Pour les PME et ETI françaises sans SOC interne, le modèle **MSSP avec Azure Lighthouse** permet à un prestataire de sécurité managé d'accéder au Sentinel du client en lecture/écriture depuis son propre tenant Azure, sans que les logs quittent l'infrastructure du client. Ce modèle de "SOC-as-a-Service délégué" est particulièrement adapté aux entités importantes NIS 2 (seuil CA > 10M€, >50 employés) qui n'ont pas les moyens d'un SOC 24/7 interne mais doivent justifier d'une capacité de détection documentée face à l'ANSSI.

La combinaison Sentinel + MDI + Azure Lighthouse couvre les 10 mesures techniques de l'article 21 NIS 2 : authentification, contrôle d'accès, chiffrement, gestion des incidents, continuité d'activité, sécurité de la chaîne d'approvisionnement, acquisition sécurisée, efficacité des mesures, hygiène cyber et formation.

Questions fréquentes sur Microsoft Sentinel et Active Directory

Quel est le coût mensuel de Microsoft Sentinel pour surveiller un Active Directory avec 3 contrôleurs de domaine ?

Sans filtrage, un contrôleur de domaine génère entre 2 et 5 Go de logs de sécurité par jour. Pour 3 DC, comptez 6 à 15 Go/jour, soit 180 à 450 Go/mois. Au tarif Pay-as-you-go d'environ 2,46 € par Go ingéré (zone Europe Ouest), cela représente 440 à 1 100 € par mois. Avec des DCR correctement configurées pour filtrer les événements à faible valeur (4634, 4648 routines), vous pouvez réduire ce volume de 40 à 60 %, ramenant le coût à 175–660 €/mois. L'engagement annuel (Commitment Tiers) réduit encore le tarif de 15 à 65 % selon le volume.

Quelle est la différence entre Microsoft Defender for Identity (MDI) et Microsoft Sentinel pour la détection AD ?

MDI est une solution spécialisée qui analyse le trafic réseau et les événements LDAP/Kerberos directement depuis les capteurs installés sur les DC, offrant une détection comportementale

temps réel sur les attaques AD (DCSync, Pass-the-Hash, Golden Ticket). Sentinel est un SIEM cloud généraliste qui corrèle des logs provenant de multiples sources. La complémentarité est clé : MDI génère des alertes de haute fidélité sur les attaques AD que Sentinel ingère via le connecteur MDI, permettant de les corréler avec d'autres signaux. MDI est inclus dans Microsoft 365 Defender P2, tandis que Sentinel facture l'ingestion.

Comment détecter un DCSync sans avoir accès aux logs réseau, uniquement via les Event ID Windows ?

Le DCSync peut être détecté via l'Event ID 4662 sur le DC, qui se déclenche quand un compte demande les droits de réplication DS-Replication-Get-Changes et DS-Replication-Get-Changes-All. La KQL dans Sentinel filtre sur les accès aux ObjectType correspondant à domainDNS et les GUIDs de réplication. Un indicateur clé est l'absence de compte machine ($) dans le SubjectUserName lors du déclenchement, ce qui différencie une réplication légitime d'un outil comme Mimikatz. MDI détecte directement cette attaque via l'analyse du trafic DRSUAPI.

Quels Event IDs Active Directory sont indispensables à collecter dans Sentinel pour couvrir le MITRE ATT&CK ?

Les Event IDs prioritaires (Tier 0) sont : 4624/4625 (authentifications), 4662 (opérations objets AD - DCSync), 4663 (accès fichiers - NTDS.dit), 4720/4726 (création/suppression comptes), 4728/4729 (membres groupes privilégiés), 4768/4769/4771 (Kerberos TGT/TGS/pré-auth), 4776 (NTLM auth), 5136 (modification objets AD - AdminSDHolder), 7045 (service installé - PsExec). Évitez 4634 (logoff), 4648 (explicit credentials routines) et 5145 (partage réseau) sans filtrage : ils génèrent un volume disproportionné.

Comment intégrer les alertes Microsoft Sentinel dans un processus de conformité NIS 2 pour une PME française ?

La directive NIS 2 impose aux entités essentielles et importantes un signalement des incidents significatifs à l'ANSSI sous 24h. Sentinel facilite cette conformité via : les Analytic Rules mappées MITRE ATT&CK qui constituent une preuve de capacité de détection, les Incidents avec timeline et entités qui alimentent le rapport d'incident ANSSI, et le connecteur Threat Intelligence TAXII pointant vers le flux CERT-FR pour bloquer les IoC connus. Pour les PME sans SOC interne, le modèle MSSP avec Sentinel délégué (Azure Lighthouse) permet d'externaliser la supervision à moindre coût tout en conservant la souveraineté des données en région France.

À RETENIR

À retenir

Les **Data Collection Rules (DCR) avec filtrage XPath** sur les Event IDs critiques AD (4662, 4769, 4768, 4728, 4663, 5136) réduisent les coûts d'ingestion Sentinel de 40 à 60 % sans perte de visibilité sur les attaques AD critiques.

La combinaison **Sentinel + Microsoft Defender for Identity** est l'architecture de référence : MDI apporte la détection comportementale réseau (DCSync, Golden Ticket, Kerberoasting) tandis que Sentinel corrèle avec les événements endpoint, Azure AD et Threat Intelligence.

Les **Watchlists "PrivilegedAccounts" et "HoneyAccounts"** enrichissent automatiquement toutes les requêtes KQL via la fonction `_GetWatchlist()`, permettant de contextualiser les alertes et de détecter immédiatement tout accès aux comptes appâts.

Les **Playbooks Logic Apps** permettent une réponse automatisée sous 60 secondes (désactivation compte, notification Teams, enrichissement TI) sans intervention

humaine pour les incidents de sévérité Critical, réduisant le MTTR de plusieurs heures à moins de 5 minutes.

Pour la **conformité NIS 2** des PME françaises, le modèle MSSP Azure Lighthouse avec Sentinel délégué en région France Central est la solution optimale : souveraineté des données garantie, capacité de détection documentée pour l'ANSSI, et coûts maîtrisés via les Commitment Tiers.

Références et documentation

Microsoft Learn — Sécurité Windows Server

ANSSI — Guide de sécurisation Active Directory

MITRE ATT&CK — Techniques Active Directory

