IEM HYBRIDE

OPEN SOURCE

WAZUH + GRAYLOG + SURICATA

Détecter, corréler et répondre
aux menaces avec des outils open source

wazuh
DÉTECTION & RÉPONSE
DES ENDPOINTS

graylog
COLLECTE, CORRÉLATION
& ANALYSE DES LOGS

SURICATA
DÉTECTION D'INTRUSION
RÉSEAU (NIDS)

Ayi NEDJIMI

VISIBILITÉ GLOBALE
DÉTECTION AVANCÉE
RÉPONSE EFFICACE
100% OPEN SOURCE

PRENEZ LE CONTRÔLE DE VOTRE SÉCURITÉ

IEM HYBRIDE OPEN SOURCE
WAZUH + GRAYLOG + SURICATA

Contents

SIEM Hybride Open Source : Wazuh + Graylog + Suricata

1. Introduction : la réalité des environnements hybrides sous menace permanente

1.1 Contexte : la menace dans les environnements hybrides AD + M365

Le déploiement d'un SIEM (Security Information and Event Management) hybride open source combinant **Wazuh**, **Graylog** et **Suricata** représente aujourd'hui l'une des architectures les plus robustes et économiquement viables pour sécuriser les environnements modernes mêlant **Active Directory on-premise** et **Microsoft 365**. Ce guide opérationnel de plus de 30 000 mots détaille l'intégralité du processus, depuis l'architecture cible jusqu'à l'opérationnalisation d'un Security Operations Center (SOC) capable de détecter et de répondre aux attaques sophistiquées modernes — Kerberoasting, DCSync, Golden Ticket, illicit OAuth consent grants, MFA fatigue, ransomwares et menaces internes. Vous découvrirez la configuration fine des trois composants, les règles de corrélation cross-domain entre AD et Microsoft 365, les runbooks SOC opérationnels, les études de cas réelles inspirées des incidents 2024-2025 documentés par Microsoft DART et Mandiant M-Trends, ainsi que les annexes complètes (règles Wazuh, Sigma, Suricata, runbooks, glossaire).

Les données de terrain 2024-2025 sont sans équivoque. Le rapport **Verizon Data Breach Investigations Report (DBIR) 2024** recense que **74 % des incidents de sécurité impliquent une composante humaine** : phishing, réutilisation de credentials, abus de privilèges. Mais au-delà de cette statistique souvent citée, c'est la corrélation entre vecteur initial et mouvement latéral qui révèle la complexité des menaces hybrides : dans 68 % des incidents analysés par Verizon, l'accès initial a été obtenu via des identifiants volés ou du phishing, puis l'attaquant a pivoté du cloud vers l'on-premises (ou inversement) en exploitant précisément les mécanismes de synchronisation d'identité.

L'équipe **Microsoft DART (Detection and Response Team)**, qui intervient sur les incidents majeurs touchant les clients Microsoft, a publié dans ses *Threat Intelligence Reports* 2024 une donnée particulièrement préoccupante : les compromissions **cloud-to-on-premise** ont augmenté de **47 % entre 2023 et 2024**. Le scénario typique commence par une compromission de compte M365 via phishing ou password spray, suivi de l'exfiltration du Primary Refresh Token (PRT) — ce jeton longue durée qui permet à un appareil Entra ID joint de s'authentifier sans MFA — puis d'une élévation de privilèges vers l'Active Directory on-premises via Azure AD Connect. Ce vecteur, connu sous le nom d'attaque **DCSync via AD Connect**, est tristement efficace car invisible pour les outils de monitoring réseau classiques.

Le rapport **Mandiant M-Trends 2024** apporte une perspective complémentaire sur le dwell time, c'est-à-dire la durée médiane de présence d'un attaquant dans le SI avant détection. Après des années de baisse régulière, ce chiffre stagne autour de **10 jours en médiane globale**, mais grimpe à **26 jours dans les environnements hybrides** où la corrélation entre événements cloud et on-premises est insuffisante. Cette asymétrie illustre précisément le problème que ce guide entend résoudre : un SIEM qui ne voit que la moitié du SI détectera trop tard, voire jamais.

Les vecteurs d'attaque spécifiques aux environnements hybrides AD + M365 méritent une analyse détaillée :

- **Azure AD Connect et ses modes de synchronisation** : le Password Hash Synchronization (PHS)

stocke des hashes de mots de passe dans Entra ID, exposant potentiellement les credentials on-premises si le tenant est compromis. Le Pass-Through Authentication (PTA) installe un agent sur les DC qui valide les authentifications en temps réel — cet agent, s'il est compromis, devient un point de pivot idéal. Le Seamless SSO crée un compte AZUREADSSOACC dans l'AD local avec un secret Kerberos partagé avec Azure : si ce secret est exfiltré (via DCSync ou dump LSASS), un attaquant peut forger des tickets Kerberos valides pour n'importe quel utilisateur.

- **Primary Refresh Tokens (PRT)** : ces tokens, émis par Entra ID pour les appareils joints, sont stockés dans le TPM sur les machines modernes, mais restent accessibles via des techniques comme `roadtx` ou des failles dans l'implémentation CloudAP. Le vol de PRT permet une authentification transparente à tous les services M365 sans déclencher de MFA.
- **Illicit Consent Grants** : technique d'attaque où un utilisateur est trompé pour accorder des permissions OAuth à une application malveillante tierce, qui obtient ainsi un accès persistant aux données M365 (mails, fichiers SharePoint, contacts) sans nécessiter de réauthentification. Mandiant a documenté plusieurs groupes APT utilisant cette technique comme vecteur de persistance furtif.

Face à ces menaces, les vulnérabilités récentes illustrent l'urgence d'une surveillance proactive. **CVE-2023-23397** (Outlook for Windows, score CVSS 9.8) permettait une authentification NTLM forcée zero-click via une simple invitation de calendrier, exposant le hash NTLMv2 des utilisateurs. **CVE-2024-21413** (Microsoft Outlook, CVSS 9.8) contournait les protections de zone de sécurité Office pour déclencher le même mécanisme via un hyperlien malformé dans un mail. Ces CVE, exploitées activement in-the-wild avant les patches, démontrent que la surface d'attaque Microsoft est permanente et critique.

Points clés à retenir :

- 74 % des incidents impliquent une composante humaine (Verizon DBIR 2024) ; 47 % d'augmentation des compromissions cloud-to-on-premises (Microsoft DART 2024).
- Le dwell time dans les environnements hybrides sans corrélation inter-couches atteint 26 jours (Mandiant M-Trends 2024) contre 10 jours en médiane globale.
- Azure AD Connect (PHS/PTA/Seamless SSO), PRT et Illicit Consent Grants sont les vecteurs hybrides les plus exploités par les groupes APT.
- CVE-2023-23397 et CVE-2024-21413 illustrent la criticité permanente des vulnérabilités Microsoft dans les environnements hybrides.

1.2 Pourquoi combiner Wazuh, Graylog et Suricata ?

La réponse instinctive de nombreuses DSI face à ces menaces est de se tourner vers des solutions propriétaires intégrées : Microsoft Sentinel, Splunk SIEM, IBM QRadar, ou Exabeam. Ces plateformes ont des qualités réelles — intégration native avec l'écosystème éditeur, support commercial, interfaces soignées. Mais leurs inconvénients structurels sont aujourd'hui rédhibitoires pour une partie croissante des organisations :

Approfondissement technique

Le coût est prohibitif à l'échelle. Microsoft Sentinel facture selon le volume de données ingérées, avec des tarifs qui oscillent entre **2,46 $ et 4,30 $ par gigaoctet** selon la région et l'engagement. Un SOC ingérant

500 Go/jour — chiffre courant pour une ETI avec 2 000 endpoints — déboursera entre 37 000 € et 65 000 € par mois en ingestion seule, sans compter les connecteurs, les licences de rétention étendues et le compute. Splunk, dont la tarification a évolué vers un modèle workload-based, reste dans des ordres de grandeur comparables, avec des contrats annuels débutant à 150 000 € pour des déploiements mid-market. Ces coûts ne sont pas justifiables pour la grande majorité des entreprises françaises de taille intermédiaire.

Le verrouillage éditeur crée une dépendance stratégique. Migrer de Splunk vers un autre SIEM après 5 ans d'utilisation implique de ré-écrire des milliers de requêtes SPL, de reconstruire tous les dashboards, de reformer les équipes. Ce vendor lock-in n'est pas accidentel : il est architectural. Les formats propriétaires, les langages de requête exclusifs (SPL, KQL, AQL) et les connecteurs certifiés créent des barrières à la sortie considérables.

La souveraineté des données est une contrainte réglementaire croissante. Le règlement **RGPD** et l'arrêt **Schrems II** (CJUE, juillet 2020) ont invalidé le Privacy Shield et rendu problématique tout transfert de données personnelles vers des serveurs américains sans garanties contractuelles solides. Or, les SIEM SaaS américains traitent par nature des logs qui peuvent contenir des données personnelles (adresses IP, noms d'utilisateurs, adresses mail). L'**ANSSI**, dans ses guides de sécurité pour les OIV et les entités soumises à NIS2, recommande explicitement d'évaluer la souveraineté de la chaîne de traitement des données de sécurité. Les solutions open source auto-hébergées répondent naturellement à cette contrainte.

La combinaison **Wazuh + Graylog + Suricata** n'est pas un assemblage arbitraire d'outils gratuits : c'est une architecture complémentaire couvrant trois couches distinctes de la pyramide de sécurité définie par le **NIST Cybersecurity Framework (CSF)** dans sa version 2.0 :

- **Wazuh** couvre la couche *endpoint et identité* : agent léger déployé sur chaque machine, il remonte les événements système (logs Windows Event Log, journald, syslog), surveille l'intégrité des fichiers (FIM), détecte les processus suspects, vérifie les vulnérabilités et intègre les logs cloud (Azure Activity, AWS CloudTrail). C'est l'œil dans chaque endpoint du SI.
- **Suricata** couvre la couche *réseau* : positionné sur un miroir de port ou en inline, il analyse le trafic en temps réel, détecte les signatures connues (Emerging Threats Open Rules, règles personnalisées), décode les protocoles applicatifs (HTTP/2, TLS 1.3, QUIC, SMB3, DNS) et peut extraire des fichiers pour analyse sandbox. C'est l'œil sur le réseau.
- **Graylog** est le *cerveau de corrélation* : il agrège les flux de Wazuh et Suricata (ainsi que tout autre source : pare-feu, équipements réseau, M365 via API, Azure Audit Logs), normalise les données dans des pipelines structurés, corrèle les événements multi-sources, et génère des alertes contextualisées. C'est là que deux événements distincts — un échec Kerberos sur un DC (Wazuh) et une connexion SMB vers une IP externe (Suricata) — deviennent une alerte Pass-the-Ticket corrélée.

Pris isolément, chaque outil présente des lacunes structurelles. **Wazuh seul** ne voit pas le réseau : il ignore les communications entre machines qui ne passent pas par un endpoint monitoré, les connexions C2 via protocoles chiffrés détectés uniquement par JA3/JA4 fingerprinting, ou les scans latéraux. **Suricata seul** manque de contexte endpoint : il voit une connexion SMB anormale mais ignore si le processus source est `lsass.exe` ou `explorer.exe`, ce qui est pourtant décisif pour qualifier l'alerte. **Graylog seul** est un moteur d'ingestion et de corrélation exceptionnel, mais sans règles de détection riches sur les endpoints et le réseau, il ne corrèle que du bruit.

L'aspect **open source auditabilité** mérite d'être développé au-delà du simple argument de gratuité.

Lorsque la CVE d'un composant propriétaire est découverte par un chercheur externe, l'éditeur choisit parfois de ne pas la publier immédiatement — notamment si elle touche à des mécanismes internes de son SIEM. Avec des logiciels open source dont le code est public sur GitHub, la communauté peut identifier, analyser et patcher les vulnérabilités de manière indépendante. Pour un outil qui est lui-même chargé de détecter les compromissions, cette transparence n'est pas anecdotique.

Points clés à retenir :

- Les SIEM propriétaires (Sentinel, Splunk) coûtent 150 €+ par Go/mois et créent un vendor lock-in stratégique incompatible avec les contraintes RGPD/Schrems II.
- Wazuh (endpoint), Suricata (réseau), Graylog (corrélation) forment un triptyque complémentaire aligné sur le NIST CSF 2.0.
- Chaque outil est insuffisant seul : la corrélation cross-couches est la valeur ajoutée architecturale centrale.
- L'open source garantit l'auditabilité du code de sécurité lui-même, critère crucial pour la souveraineté et la conformité ANSSI.

1.3 Objectifs et périmètre de ce guide

Ce guide a pour ambition de documenter, avec le niveau de détail opérationnel nécessaire à une mise en production réelle, le déploiement et la configuration d'un **SIEM hybride open source** couvrant les environnements Active Directory on-premises et Microsoft 365. Il ne s'agit pas d'une introduction théorique : chaque section fournit les commandes, configurations et arbitrages nécessaires à un ingénieur sécurité expérimenté pour construire un SOC fonctionnel.

Le périmètre couvert dans ce guide complet inclut le déploiement de l'infrastructure SIEM (cette partie 1), l'intégration Active Directory et M365 (partie 2), la détection avancée des menaces (TTPs MITRE ATT&CK, règles Sigma, règles Suricata custom), la gestion des incidents (playbooks, enrichissement IOC, intégration TheHive/Cortex), l'optimisation et la haute disponibilité, la conformité (NIS2, ISO 27001, RGPD), et la maintenance opérationnelle long terme.

1.4 Public cible et prérequis

Ce guide s'adresse aux **ingénieurs et architectes sécurité** avec une expérience opérationnelle préalable. Les prérequis techniques incluent : maîtrise de Linux (Ubuntu 24.04 LTS, RHEL/Rocky), connaissance de l'écosystème Active Directory (GPO, Kerberos, LDAP, NTLM), compréhension des fondamentaux réseau (TCP/IP, VLAN, SPAN/TAP), notions de Docker et d'orchestration de base. Une expérience préalable avec un SIEM quel qu'il soit est recommandée mais non obligatoire.

1.5 Méthodologie et environnement de test

L'environnement de validation de ce guide repose sur un cluster **Proxmox VE 8.2** hébergeant un lab complet simulant une infrastructure d'ETI type : deux contrôleurs de domaine Windows Server 2022 en réplication, un tenant Microsoft 365 Business Premium (trial), une DMZ avec pare-feu OPNsense, des postes clients

Windows 11 23H2, des serveurs Linux Ubuntu 24.04 LTS, et l'infrastructure SIEM elle-même. La génération d'attaques réalistes s'appuie sur trois frameworks complémentaires : **Atomic Red Team** (Invoke-AtomicTest, bibliothèque de 900+ techniques MITRE), **MITRE Caldera** (agents C2 simulés, campagnes multi-étapes automatisées), et **BloodHound/SharpHound** (énumération AD, identification des chemins d'attaque vers Domain Admin). Cette combinaison permet de valider les règles de détection contre des TTPs réalistes sans compromettre un SI de production.

---

2. Présentation détaillée des technologies

2.1 Wazuh : SIEM/XDR open source nouvelle génération

2.1.1 Architecture technique approfondie **Wazuh** est né en 2015 comme fork de **OSSEC**, le célèbre HIDS open source créé par Daniel Cid, pour en corriger les limitations architecturales et y ajouter une dimension SIEM complète. En 2025, la version **4.10+** positionne Wazuh comme une plateforme XDR (Extended Detection and Response) mature, avec une architecture en quatre composants distincts :

Le Wazuh Agent est un processus léger écrit en C, occupant environ **5 à 15 Mo de mémoire RAM** au repos et **1 à 3 % de CPU** sur des charges normales. Il est disponible pour Windows (XP à Server 2025), Linux (kernel 3.x+), macOS, FreeBSD, Solaris, HP-UX et AIX — une couverture multiplateforme que peu de solutions commerciales égalent. L'agent collecte plusieurs catégories d'événements : journaux système (Windows Event Log via WEL API, journald, syslog), changements de fichiers (FIM), inventaire des processus actifs, états des ports réseau, packages installés, et peut exécuter des commandes d'audit personnalisées (module `wodle command`). Il transmet ces données chiffrées au Wazuh Manager.

Le Wazuh Manager (anciennement Wazuh Server) est le composant central de traitement. Il reçoit les événements des agents via le port **1514/UDP ou TCP** (configurable, TLS recommandé en production) et le port **1515/TCP** pour l'enrôlement des agents avec authentification par certificat. Le protocole de communication implémente un chiffrement **AES-256-CBC** avec échange de clés au moment de l'enrôlement. Le Manager applique successivement les **décodeurs** (parsing des formats de logs) puis le **moteur de règles** (évaluation des conditions de détection, fréquence, corrélation temporelle) pour générer des alertes. En version 4.10, le moteur de règles intègre plus de **3 200 règles** couvrant les attaques Windows, Linux, cloud, réseau et applications.

Le Wazuh Indexer est un fork de **OpenSearch 2.x** (lui-même fork d'Elasticsearch 7.10 avant le changement de licence Elastic vers SSPL en 2021). Il stocke les alertes et les logs bruts dans des index rotatifs avec une politique de rétention configurable (ILM — Index Lifecycle Management). L'API REST compatible avec Elasticsearch 7.10 permet l'intégration avec des outils tiers comme Graylog, que nous verrons en détail dans les sections suivantes.

Le Wazuh Dashboard est un fork de **OpenSearch Dashboards** (lui-même fork de Kibana 7.10), enrichi de tableaux de bord dédiés sécurité : agents overview, alertes MITRE ATT&CK, conformité PCI/HIPAA/GDPR, FIM, vulnérabilités. En version 4.10, l'interface intègre un module de **Threat Intelligence** avec enrichissement MISP et VirusTotal natif.

La distinction entre déploiement **All-in-One** et **distribué** est critique pour le dimensionnement. En All-in-One, Manager + Indexer + Dashboard coexistent sur un seul nœud — acceptable jusqu'à environ **500 agents** avec du matériel adapté (16 vCPU, 32 Go RAM, SSD NVMe). Au-delà, l'architecture distribuée devient nécessaire : l'Indexer passe en cluster multi-nœuds avec sharding (minimum 3 nœuds pour la haute disponibilité), les Managers peuvent être configurés en cluster actif-actif avec synchronisation des règles, et le Dashboard devient un nœud dédié. La scalabilité horizontale de l'Indexer (OpenSearch) permet théoriquement de gérer plusieurs dizaines de milliers d'agents, mais les performances réelles dépendent fortement du volume d'événements par agent et de la complexité des règles de corrélation.

Table 1: Wazuh : comparaison déploiement All-in-One vs Distribué

Critère	All-in-One	Distribué (minimum)	Distribué (production)
Agents supportés	Jusqu'à 500	500 à 3 000	3 000 à 50 000+
Nœuds minimum	1	4 (1 manager, 3 indexer)	7+ (cluster manager + cluster indexer)
RAM totale recommandée	32 Go	64 Go	128 Go+
Stockage (1 an, 500 agents)	2 To SSD	6 To SSD (répliqué)	Selon rétention + tiering S3
SPOF	Total	Manager (si 1 seul)	Aucun avec HA
Complexité opérationnelle	Faible	Moyenne	Élevée

2.1.2 Fonctionnalités phares de Wazuh 4.10 Le module **EDR (Endpoint Detection and Response)** de Wazuh va bien au-delà de la simple collecte de logs. Le module `syscollector` maintient un inventaire en temps réel des processus actifs (avec leur arbre de processus parent-enfant, utile pour détecter `cmd.exe` spawné par `winword.exe`), des connexions réseau ouvertes (ports locaux, IPs distantes, PID associé), et des clés de registre Windows critiques. Cette visibilité process-level est ce qui distingue un vrai XDR d'un simple collecteur de logs.

Considérations opérationnelles

Le File Integrity Monitoring (FIM) utilise **Auditpol** sur Windows (API Windows Audit Policy) et **inotify/fanotify** sur Linux pour une surveillance en temps réel (plutôt que polling) des modifications de fichiers, répertoires et clés de registre. Les chemins surveillés sont configurables par politique, avec des listes d'exclusion pour éviter le bruit des fichiers de log applicatifs. Le FIM génère des événements contenant le hash SHA-256 avant/après modification, l'utilisateur responsable et le processus modifiant — données cruciales pour la forensique et la réponse à incident.

La Vulnerability Detection (module SCA — Security Configuration Assessment et module `vulnerability-detector`) agrège plusieurs sources de données CVE : le **National Vulnerability Database (NVD)** du NIST, les bulletins **Microsoft Update Catalog** (MSRC), les flux **Canonical OVAL** pour Ubuntu/Debian, et les advisory Red Hat. Le croisement avec l'inventaire des packages installés sur chaque agent génère des alertes de vulnérabilité contextualisées avec le score CVSS, les versions affectées et les

références aux patches disponibles. En version 4.10, la détection couvre également les packages Python (pip), npm et gem via le module `syscollector` étendu.

La **détection de rootkits** (module `rootcheck`) vérifie des indicateurs de compromission bas niveau : fichiers cachés (discordance entre listing système et listing raw du système de fichiers), processus invisibles (comparaison /proc avec liste kernel), ports cachés, et binaires système modifiés (comparaison avec une baseline de hashes). Ces vérifications sont coûteuses en CPU et sont configurées pour tourner en périodique (toutes les 12h par défaut) plutôt qu'en temps réel.

L'intégration **cloud native** de Wazuh 4.10 est particulièrement pertinente pour les environnements hybrides : le module `azure-logs` collecte les **Azure Activity Logs**, les **Microsoft Entra ID Sign-in Logs** et les **Microsoft 365 Audit Logs** via l'API Microsoft Graph, sans agent supplémentaire. De même, `aws-s3` collecte CloudTrail, VPC Flow Logs et GuardDuty findings depuis S3. Ces intégrations cloud complètent la visibilité endpoint pour couvrir le SI hybride dans sa totalité.

Points clés à retenir :

- Wazuh Agent : ~5-15 Mo RAM, 1-3 % CPU, disponible sur toutes plateformes (Windows XP → Server 2025, Linux, macOS, BSD).
- Communication agent-manager : AES-256-CBC, ports 1514 (données) et 1515 (enrôlement), TLS obligatoire en production.
- All-in-One jusqu'à ~500 agents (32 Go RAM, SSD) ; au-delà, architecture distribuée avec cluster OpenSearch minimum 3 nœuds.
- FIM temps réel via inotify/fanotify (Linux) et Auditpol (Windows) ; Vulnerability Detection via NVD + MSRC + OVAL ; Cloud native Azure/AWS/GCP.

2.1.3 Wazuh Indexer et la compatibilité OpenSearch Le choix d'OpenSearch comme backend de stockage n'est pas anodin. En janvier 2021, Elastic NV a changé la licence d'Elasticsearch et Kibana de l'Apache License 2.0 vers la **Server Side Public License (SSPL)** et l'Elastic License 2.0, restreignant l'usage commercial en mode SaaS. AWS a répondu en forkant la dernière version Apache 2.0 (Elasticsearch 7.10.2) pour créer **OpenSearch**, maintenu sous Apache License 2.0. Wazuh a suivi cette décision architecturale, garantissant la pérennité open source de son stack.

Cette décision a une implication pratique capitale pour notre architecture : **Graylog 6.x supporte nativement OpenSearch comme backend de stockage**, en plus d'Elasticsearch. La compatibilité API entre Wazuh Indexer (OpenSearch) et Graylog permet donc d'envisager une architecture où Graylog requête directement l'Indexer Wazuh pour certains cas d'usage, bien que l'approche recommandée dans ce guide soit de transiter par Filebeat pour l'envoi vers Graylog.

2.2 Graylog 6.x : la plateforme de gestion des logs centralisée

2.2.1 Architecture interne de Graylog **Graylog**, développé par la société allemande Graylog Inc. (anciennement TORCH GmbH), est en version **6.1+** en 2025. Son architecture repose sur trois composants fondamentaux qui peuvent être déployés séparément pour la scalabilité :

Le Graylog Server est une application **JVM** (Java Virtual Machine) écrite principalement en Java, avec des composants frontend en JavaScript/React. Il orchestre l'ensemble : réception des logs entrants (Inputs), traitement via les pipelines (Processors), routage vers le stockage, évaluation des règles d'alerte, et exposition de l'interface web et de l'API REST. La version 6.1 requiert **Java 17 LTS** (OpenJDK). La JVM est un facteur de tuning critique : une heap mal dimensionnée provoque des GC pauses qui créent des pertes de logs en période de forte charge.

Le MongoDB est utilisé exclusivement pour les **métadonnées** de Graylog : configuration des Inputs, définitions des Streams, règles d'alerte, utilisateurs et rôles RBAC, paramètres de pipelines. MongoDB ne stocke jamais les messages de logs eux-mêmes. La version 6.x de Graylog requiert MongoDB 5.0+ et recommande MongoDB 6.0 ou 7.0 pour les performances. En configuration haute disponibilité, un **Replica Set MongoDB** à 3 membres est obligatoire pour garantir le quorum.

OpenSearch (ou Elasticsearch) est le backend de stockage des messages eux-mêmes. Chaque message reçu par Graylog est indexé dans OpenSearch avec ses champs (timestamp, source, niveau de sévérité, champs extraits par les pipelines) et rendu searchable quasi-instantanément. Graylog gère automatiquement la rotation des indices selon le volume ou le temps, et supporte les **Index Sets** permettant d'appliquer des politiques de rétention différentes par type de données (logs réseau conservés 30 jours, alertes sécurité 1 an, conformité 5 ans).

2.2.2 Forces distinctives de Graylog La **capacité d'ingestion** de Graylog est l'une de ses forces majeures. Il supporte nativement une vingtaine de protocoles d'entrée : GELF (Graylog Extended Log Format) via TCP/UDP/HTTP (format JSON natif Graylog), **Syslog** RFC 3164 et 5424, **Beats** (Filebeat, Winlogbeat, Packetbeat), **Kafka** consumer (intégration avec les pipelines streaming), **AWS Kinesis**, **CEF (Common Event Format)** utilisé par de nombreux équipements réseau et pare-feu. Cette polyvalence permet d'ingérer des sources hétérogènes sans agent intermédiaire dédié.

Détails de mise en œuvre

Le **Pipeline Processing** est l'élément différenciateur de Graylog par rapport à une stack ELK pure. Un pipeline est une série de **stages** (étapes ordonnées) contenant des **règles** écrites en Graylog Processing Language (GPL) — un DSL fonctionnel. Ces règles permettent de parser, transformer, enrichir et router les messages : extraire des champs depuis une regex, résoudre une IP en géolocalisation, tagger un message selon sa sévérité, supprimer des champs sensibles avant stockage (pseudonymisation RGPD), ou router vers un stream spécifique selon des critères complexes. Cette approche est plus structurée et maintenable que les filtres Logstash (grok + conditionnels imbriqués) pour des volumes importants.

Le **système d'alerting** Graylog 6.x est basé sur les Event Definitions : des conditions qui s'évaluent sur les streams de logs en temps réel ou en agrégation temporelle. Les notifications supportent nativement Slack, Teams, PagerDuty, e-mail, webhooks HTTP et — critique pour notre architecture — **TheHive** via plugin. La gestion des **suppressions** (ne pas alerter deux fois pour le même incident) et des **corrélations temporelles** (alerter si X événements de type A suivis de Y événements de type B dans une fenêtre de Z minutes) couvre la majorité des besoins SOC.

La comparaison avec **ELK (Elasticsearch-Logstash-Kibana)** mérite nuance. ELK offre plus de flexibilité dans les pipelines Logstash et une meilleure intégration avec l'écosystème Elastic (APM, Fleet, Elastic Security). Mais Graylog a une gestion des streams et de la multi-tenancy plus élaborée, une interface orientée équipes SOC (pas seulement développeurs), et une gestion des rôles RBAC plus fine. Pour un SOC axé sur la sécurité plutôt que sur l'observabilité applicative, Graylog est généralement le meilleur choix. La comparaison avec **Splunk** tourne en faveur de Graylog sur le TCO (Total Cost of Ownership) — un déploiement Graylog ingérant 500 Go/jour coûte en infrastructure (serveurs ou cloud) environ 3 000 à 8 000 € par mois, soit 10 à 20 fois moins que Splunk.

Points clés à retenir :

- Graylog 6.x : Graylog Server (JVM Java 17) + MongoDB (métadonnées uniquement) + OpenSearch (stockage messages).
- Ingestion multi-protocole native : GELF, Syslog, Beats, Kafka, CEF, AWS Kinesis — aucun agent intermédiaire requis pour la plupart des sources.
- Pipeline Processing GPL : parsing, enrichissement, pseudonymisation, routage — plus structuré que Logstash grok pour les besoins SOC.
- TCO 10 à 20 fois inférieur à Splunk pour des volumes comparables ; approche log-management orientée SOC vs ELK orienté observabilité.

2.2.3 Graylog comme cerveau de corrélation du SIEM hybride Dans notre architecture, Graylog joue un rôle de **normalisation et corrélation cross-sources** que ni Wazuh Indexer ni Suricata ne peuvent assurer seuls. Il reçoit les alertes Wazuh (via Filebeat depuis Wazuh Manager), les événements Eve JSON de Suricata, les logs des équipements réseau (pare-feu, switches avec 802.1X), et les API cloud (M365 Audit via Logstash ou plugin). Dans les pipelines Graylog, ces données sont normalisées vers un schéma commun (basé sur l'**Elastic Common Schema — ECS**), permettant à une règle d'alerte d'évaluer des champs cohérents quelle que soit la source.

La corrélation devient alors possible : détecter qu'un utilisateur dont le compte AD a déclenché une alerte Wazuh (échec Kerberos répété — potentielle attaque AS-REP Roasting) présente simultanément une connexion réseau vers une IP catégorisée C2 dans les alerts Suricata, ET que ce même compte a eu une connexion M365 depuis une IP géographiquement distante 30 minutes plus tôt — tout cela dans une seule alerte Graylog corrélée.

2.3 Suricata 7.0 : détection réseau nouvelle génération

2.3.1 Moteurs de détection et protocoles supportés **Suricata**, développé par l'**Open Information Security Foundation (OISF)** avec le soutien de la communauté et de DHS américain dans les années 2000-2010, est en version **7.0+** depuis 2023 avec des mises à jour régulières en 2024-2025. Il combine plusieurs moteurs de détection complémentaires qui en font bien plus qu'un simple IDS/IPS basé sur des signatures :

Le **moteur de signatures** évalue les règles en format Snort-compatible (avec extensions Suricata spécifiques) contre chaque paquet réseau. Les Emerging Threats Open Rules (ET Open), maintenues par Proofpoint, constituent la base de données de signatures open source la plus complète, avec plus de **45**

000 règles actives couvrant malwares, exploits, C2 connus, scans réseau, et anomalies protocolaires. Des jeux de règles commerciaux (ET Pro, Stamus Networks) ajoutent des règles premium avec couverture des menaces les plus récentes.

La détection de protocoles (Protocol Detection et Application Layer Parsers) est la capacité la plus distinctive de Suricata 7.0. Contrairement à un simple packet filter, Suricata comprend la sémantique applicative de plus de **50 protocoles** : HTTP/1.1 et HTTP/2 (avec décodage des headers, corps de requête, User-Agent, méthodes anormales), **TLS 1.2 et 1.3** (analyse des certificats, JA3/JA4 fingerprinting des clients, détection de TLS auto-signé ou expiré), **QUIC** (UDP 443, de plus en plus utilisé par les C2 modernes), **SMB 2/3** (détection des patterns WannaCry, EternalBlue, accès massivement parallèles évocateurs de ransomware), **DNS** (DGA detection, tunneling, requêtes vers des domaines malveillants), **RDP, SSH, FTP, SMTP, IMAP**. Cette compréhension protocolaire permet des règles de détection sémantiquement riches, impossibles avec une analyse de paquets bruts.

La **détection d'anomalies** s'appuie sur des modèles statistiques et des heuristiques : dépassement de fréquence de connexions (rate limiting), volumes de données anormaux, comportements protocolaires non-conformes aux RFC. En version 7.0, Suricata intègre une détection d'anomalie plus fine sur les flux TLS (certificats auto-signés, longueur de Session-ID anormale, suites cryptographiques désuètes).

L'**extraction de fichiers** (`filestore`) permet à Suricata d'extraire les fichiers transitant en clair sur le réseau (HTTP, FTP, SMB sans chiffrement) pour envoi vers un sandbox (Cuckoo, CAPE, Any.run via API). Cette capacité est précieuse pour l'analyse de malwares distribués via partages réseau ou téléchargements HTTP internes.

Aspects pratiques

2.3.2 Eve JSON : le format de sortie universel La sortie Eve JSON est l'un des choix architecturaux les plus judicieux de Suricata : tous les événements (alertes, flux réseau, métadonnées HTTP/DNS/TLS, fichiers extraits) sont emis dans un fichier `/var/log/suricata/eve.json` au format JSON structuré, un enregistrement par ligne. Chaque événement contient des champs cohérents : `timestamp` (ISO 8601 avec microseconde), `event_type` (alert, dns, http, tls, flow, fileinfo), `src_ip`, `dest_ip`, `proto`, et des champs spécifiques au type d'événement. Ce format est nativement parsé par Filebeat (module Suricata intégré), Logstash, Fluentd, et directement par Graylog via un Input JSON line par line.

Pour la détection des C2 modernes utilisant TLS, le champ `tls.ja3` dans les events Eve JSON de type `tls` est particulièrement précieux. Le JA3 fingerprint est un hash MD5 dérivé des paramètres du ClientHello TLS (version, suites cryptographiques, extensions, courbes elliptiques) — il identifie le client TLS de manière quasi-unique, indépendamment de l'IP ou du domaine de destination. Des bibliothèques de JA3 malveillants connus (Abuse.ch, JARM) permettent de détecter des clients C2 même lorsqu'ils changent d'infrastructure. Suricata 7.0 supporte également **JA4**, successeur de JA3 avec une meilleure robustesse aux variations mineures d'implémentation.

2.3.3 Modes de déploiement IDS vs IPS Suricata peut opérer en deux modes fondamentalement différents, avec des implications architecturales majeures :

En mode **IDS (Intrusion Detection System)**, Suricata analyse une copie du trafic réseau via un port SPAN (Switch Port ANalyzer) ou un TAP réseau physique/optique. Il ne peut que détecter et alerter — il ne peut pas bloquer le trafic. C'est le mode recommandé pour le déploiement initial (zéro impact sur le trafic de production) et pour les réseaux où le mode inline est impraticable (liens fibre haute vitesse, topologies complexes). Le trafic dupliqué est injecté dans l'interface de capture Suricata via **AF_PACKET** (mode kernel bypass à haute performance, recommandé sur Linux) ou **PF_RING** (option encore plus performante pour des débits > 10 Gbps).

En mode **IPS (Intrusion Prevention System)**, Suricata est positionné en coupure sur le chemin du trafic via **NFQUEUE** (Netfilter Queue, intégration iptables/nftables sur Linux). Chaque paquet est mis en attente dans le kernel et remis à Suricata pour évaluation avant transmission ou abandon. Ce mode permet le blocage en temps réel (action `drop` dans les règles) mais introduit de la latence (typiquement 0,5 à 2 ms sur du matériel moderne) et devient un point de défaillance unique si Suricata plante. En production, le mode IPS requiert une stratégie de **fail-open** (bypass hardware ou software si Suricata ne répond plus) pour éviter une coupure réseau totale.

Table 2: Suricata : comparaison modes IDS vs IPS

Critère	Mode IDS (SPAN/TAP)	Mode IPS (NFQUEUE)
Impact trafic production	Aucun (copie du trafic)	Latence 0,5-2 ms ; point de défaillance
Capacité de blocage	Non (détection uniquement)	Oui (action drop en temps réel)
Risque opérationnel	Faible	Élevé sans fail-open hardware
Performance maximale	40+ Gbps (AF_PACKET + PF_RING)	10-20 Gbps (limité par NFQUEUE)
Déploiement recommandé	Initial, réseaux haute disponibilité	DMZ, segments critiques contrôlés
Faux positifs bloquants	Impact nul (pas de blocage)	Risque d'interruption de service

Points clés à retenir :

- Suricata 7.0 : 45 000+ règles ET Open, parsers de 50+ protocoles (HTTP/2, TLS 1.3, QUIC, SMB3), JA3/JA4 fingerprinting, extraction de fichiers.
- Eve JSON : format de sortie structuré universel, nativement parsé par Graylog, Filebeat, Logstash — clé de l'intégration.
- IDS (SPAN/TAP) recommandé pour le déploiement initial : détection sans risque ; IPS (NFQUEUE) pour les segments où le blocage en temps réel est requis.
- JA4 fingerprinting TLS permet de détecter les C2 qui changent d'infrastructure mais conservent le même client malveillant.

---

3. Architecture cible d'intégration

3.1 Modèle de données et flux de logs

Avant de dessiner les flux techniques, il faut établir une **taxonomie claire des sources de données** que le SIEM hybride doit ingérer. Cette taxonomie guide les décisions d'architecture : quel collecteur pour quelle source, quel pipeline de normalisation, quelle politique de rétention.

Les sources se répartissent en quatre grandes familles :

- **Sources endpoint** : Windows Event Log (Security, System, Application, PowerShell Operational, Sysmon), journald/syslog Linux, macOS unified logging. Collectées via agent Wazuh. Volumes typiques : 50 à 500 EPS (Events Per Second) par endpoint, selon la verbosité de l'audit policy.
- **Sources réseau** : Suricata Eve JSON (alertes + métadonnées flows), NetFlow/sFlow/IPFIX des équipements réseau, logs pare-feu (OPNsense/pfSense syslog, Fortinet, Palo Alto CEF), logs proxy (Squid, ZScaler), DNS logs (Bind9, Windows DNS analytical log). Volumes : 500 à 50 000 EPS selon le débit réseau et la verbosité.
- **Sources identité** : Windows Security Event Log sur les Domain Controllers (Event IDs 4624, 4625, 4648, 4768, 4769, 4771, 4776 pour Kerberos/NTLM), Azure AD Sign-in Logs (via Microsoft Graph API), MFA logs (Duo, Entra ID MFA), PAM logs Linux. Critique pour la détection des attaques d'identité.
- **Sources cloud** : Microsoft 365 Unified Audit Log (Exchange, SharePoint, Teams, OneDrive — via Graph API ou Azure Event Hub), Azure Activity Log (opérations sur les ressources Azure), Azure Diagnostic Logs (NSG flow logs, WAF logs), AWS CloudTrail, GCP Audit Logs. Volumes variables, généralement 10 à 1 000 EPS selon l'activité utilisateur.

Les **formats de logs** rencontrés dans ce contexte hétérogène sont multiples et posent le défi de la normalisation :

Table 3: Formats de logs sources et méthodes d'ingestion Graylog

Format	Source type	Input Graylog	Parsing requis
Windows Event Log (XML/EVTX)	Windows endpoints, DCs	Beats (Winlogbeat)	Extracteur champs Wir
Syslog RFC 5424	Linux, équipements réseau	Syslog UDP/TCP	Grok ou Regex extracto
Eve JSON (Suricata)	Suricata IDS/IPS	Beats (Filebeat) ou JSON	Minimal (JSON natif)
CEF (Common Event Format)	Pare-feu, WAF, IDS commerciaux	Syslog + extracteur CEF	Parser CEF intégré
JSON (API cloud)	M365, Azure, AWS	HTTP JSON ou Kafka	Pipeline de normalisat
GELF natif	Applications, Wazuh via Filebeat	GELF UDP/TCP	Aucun (format natif)

La **normalisation vers l'Elastic Common Schema (ECS)** est la décision architecturale qui rend les corrélations cross-sources possibles. L'ECS définit un vocabulaire commun pour les champs les plus importants : `source.ip`, `destination.ip`, `user.name`, `process.name`, `event.category`, `event.type`, `event.outcome`. Lorsque les pipelines Graylog mappent chaque source vers ces champs

communs, une règle d'alerte peut comparer `user.name` entre un log Wazuh Windows et un log M365 sans se soucier du format source.

3.2 Positionnement des composants dans l'architecture physique

L'architecture réseau cible positionne les composants SIEM dans des zones de sécurité distinctes, conformément au principe de **défense en profondeur** :

La **zone SIEM** (VLAN dédié, accès restreint) héberge les serveurs Graylog, OpenSearch et MongoDB. Ces serveurs ne doivent recevoir que des flux de logs entrants (push des agents et collecteurs) — ils ne doivent jamais initier de connexions vers le SI de production, à l'exception des requêtes de collecte API (M365, Azure) qui sortent vers Internet via un proxy dédié.

Le **Wazuh Manager** peut être colocalisé dans la zone SIEM ou dans une zone de gestion séparée. Les agents Wazuh sur les endpoints de production établissent des connexions sortantes vers le Manager (port 1514/1515) — ce sens de connexion (endpoint → Manager) est important pour la politique de pare-feu : les endpoints n'acceptent pas de connexions entrantes depuis le SIEM.

Les **sondes Suricata** sont positionnées aux points de collecte réseau stratégiques : sur les ports SPAN des switches d'accès pour la visibilité interne, en entrée/sortie de la DMZ pour la visibilité Internet, et potentiellement sur les liens d'interconnexion datacenter-cloud (VPN site-to-site Azure/AWS). Chaque sonde Suricata dispose d'une interface dédiée à la capture (mode promiscuous, pas d'IP) et d'une interface de management pour l'envoi des logs Eve JSON vers Graylog.

Les **collecteurs cloud** (module azure-logs de Wazuh, ou scripts Python/Lambda dédiés) sont de préférence externalisés dans une fonction serverless ou un serveur dédié avec accès Internet contrôlé, pour éviter de router le trafic API Microsoft/AWS à travers l'infrastructure de production.

3.3 Haute disponibilité et scalabilité

Pour un SIEM de production, la haute disponibilité n'est pas optionnelle : un SIEM en panne est pire qu'un SIEM absent, car il crée une fausse impression de surveillance. L'architecture HA repose sur plusieurs mécanismes :

Pour **Graylog Server**, une configuration multi-nœuds Graylog (minimum 2 nœuds actifs-actifs) avec un load balancer **HAProxy** en amont distribue la charge d'ingestion. La coordination des nœuds Graylog utilise MongoDB comme registre de configuration partagé. Les inputs Beats sur chaque nœud Graylog acceptent les connexions : si un nœud tombe, HAProxy redirige vers le nœud restant sans perte de données (les agents Beats réessaient automatiquement).

Pour **OpenSearch**, un cluster à minimum **3 nœuds** avec des rôles distincts (master-eligible, data, coordinating) garantit qu'aucun nœud unique n'est un SPOF. Le paramètre `number_of_replicas: 1` sur les index Graylog assure qu'une copie de chaque shard est présente sur un nœud différent — si un nœud data tombe, les shards primaires restants et leurs répliques permettent un fonctionnement continu sans perte de données.

Pour **MongoDB**, un Replica Set à 3 membres (1 primary, 2 secondary) avec élection automatique du primary en cas de défaillance garantit la persistance des métadonnées Graylog. Un arbiter (membre sans données, uniquement pour le vote) peut remplacer le troisième secondary pour réduire les coûts si le volume de métadonnées est faible.

La **haute disponibilité Wazuh** repose sur un cluster Wazuh Manager (2+ nœuds) synchronisant règles et état des agents. Un VIP (Virtual IP) géré par **Keepalived/VRRP** devant les managers garantit une IP fixe pour les agents, même si le manager actif tombe. En mode cluster Wazuh, les agents maintiennent une liste de managers de failover dans leur configuration locale.

3.4 Sécurité des communications inter-composants

Le SIEM lui-même est une cible de choix pour un attaquant cherchant à désactiver la surveillance ou à exfiltrer des données de sécurité. La sécurisation des communications est donc critique :

- **mTLS (mutual TLS)** entre tous les composants : Wazuh agents ↔ Manager (TLS 1.3 avec certificats auto-signés CA interne), Graylog Server ↔ OpenSearch (TLS 1.3, certificats signés CA interne), Graylog Server ↔ MongoDB (TLS optionnel mais recommandé), Filebeat ↔ Graylog Beats Input (TLS 1.2 minimum).
- **Segmentation VLAN** : le trafic de logs ne doit pas transiter sur les VLANs de production. Un VLAN SIEM dédié avec règles de pare-feu strictes (seuls les ports de collecte sont autorisés en entrée depuis les zones de production).
- **RBAC LDAP/AD** : l'interface Graylog et le Dashboard Wazuh s'authentifient contre l'Active Directory ou LDAP via LDAP TLS, avec des groupes AD mappés aux rôles Graylog (viewer, analyst, admin). L'accès direct aux APIs OpenSearch et MongoDB est bloqué depuis l'extérieur de la zone SIEM.
- **Chiffrement au repos** : les volumes OpenSearch et MongoDB sont chiffrés via LUKS ou les mécanismes de chiffrement du cloud provider, protégeant les données en cas d'accès physique au serveur.

3.5 Stratégie de stockage hot/warm/cold

La politique de rétention des logs de sécurité doit équilibrer les contraintes réglementaires (NIS2 impose 12 mois de conservation, ISO 27001 recommande 1 à 3 ans selon la criticité), les performances de recherche (les analystes SOC interrogent principalement les 7-30 derniers jours), et le coût du stockage (SSD NVMe pour le hot tier, HDD pour le warm, objet S3/MinIO pour le cold).

L'architecture de tiering recommandée dans Graylog/OpenSearch :

- **Hot tier** (7-30 jours) : SSD NVMe, index OpenSearch actifs, latence de recherche <100ms. Toutes les sources. RAM dédiée : 50% du heap OpenSearch pour le cache de filesystem.
- **Warm tier** (30-90 jours) : HDD SATA, index OpenSearch read-only (searchable mais pas indexables), latence 500ms-2s acceptable. Recherche forensique.
- **Cold tier** (90 jours à 5 ans) : S3 ou **MinIO** (S3-compatible auto-hébergé pour la souveraineté), via la fonctionnalité Searchable Snapshots d'OpenSearch 2.x. Les données sont compressées et stockées

en objet, montées à la demande pour les recherches d'investigation. Latence : plusieurs secondes à dizaines de secondes, acceptable pour des investigations historiques.

3.6 Dimensionnement : calcul des EPS et recommandations matérielles

Le dimensionnement d'un SIEM est l'exercice le plus critique — et le plus souvent mal fait — du déploiement. Les calculs ci-dessous sont basés sur des mesures réelles en production et constituent des ordres de grandeur à affiner selon le profil spécifique de l'organisation.

La charge est mesurée en EPS (Events Per Second), le nombre d'événements traités par seconde en régime permanent, avec des pics lors d'incidents ou de scans. Les sources suivantes contribuent à l'EPS total :

Table 4: Estimation EPS par type de source (ETI 500 endpoints)

Source	Volume unitaire	Nombre	EPS estimé	EPS en pic
Wazuh agents Windows (postes)	2-5 EPS/agent	300	600-1500	5000
Wazuh agents Windows Server	5-20 EPS/serveur	50	250-1000	3000
Wazuh agents Linux	1-3 EPS/agent	100	100-300	1000
Suricata (réseau 1 Gbps)	500-5000 EPS/sonde	3 sondes	1500-15000	30000
Domain Controllers (DC logs)	50-200 EPS/DC	4	200-800	2000
M365 Unified Audit Log	10-100 EPS total	1 tenant	10-100	500
Pare-feu / équipements réseau	50-500 EPS/équipement	5	250-2500	5000
TOTAL estimé			**3000-21000 EPS**	**46500 EPS**

Sur la base d'un EPS moyen de **10 000 EPS** (ETI 500 endpoints, trafic réseau modéré), les recommandations matérielles pour chaque composant sont les suivantes :

- **Graylog Server** (2 nœuds HA) : 8 vCPU, 32 Go RAM (dont 8 Go heap JVM), 200 Go SSD (OS + journaux Graylog). La heap JVM ne doit jamais dépasser 31 Go (limite before compressed OOPs en Java). En dessous de 6 Go de heap, les GC pauses deviennent fréquentes au-delà de 5 000 EPS.
- **OpenSearch** (3 nœuds data) : 16 vCPU, 64 Go RAM (32 Go heap OpenSearch + 32 Go page cache filesystem), 4 To SSD NVMe par nœud (hot tier 30 jours à ~10 Ko/event). La règle empirique OpenSearch : heap = 50% RAM disponible, jamais plus de 31 Go.
- **MongoDB Replica Set** (3 membres) : 4 vCPU, 8 Go RAM, 100 Go SSD. MongoDB Graylog ne stocke que des métadonnées — les besoins sont faibles.
- **Wazuh Manager** (cluster 2 nœuds) : 8 vCPU, 16 Go RAM, 500 Go SSD. Le Manager est CPU-bound lors du traitement des règles à fort EPS.
- **Suricata par sonde** (IDS, 1 Gbps) : 8 vCPU (worker threads = CPU-1), 16 Go RAM (ring buffers AF_PACKET), interface 10GbE pour la capture + 1GbE management.

Points clés à retenir :

- Calculer l'EPS total avant tout déploiement : une ETI 500 endpoints génère 3 000-21 000 EPS en régime normal, avec des pics à 46 500 EPS lors d'incidents.
- Règle OpenSearch critique : heap = 50% RAM, jamais > 31 Go. Sous-dimensionner l'heap = GC pauses = pertes de logs.
- Architecture hot/warm/cold avec MinIO S3 pour le cold tier : conformité réglementaire (NIS2 = 12 mois minimum) sans coût prohibitif de SSD.
- Tout SIEM en production doit être HA : Keepalived/VRRP pour Wazuh Manager, HAProxy + cluster Graylog, OpenSearch 3 nœuds minimum.

---

4. Déploiement et configuration de base

4.1 Préparation de l'infrastructure

Avant d'installer le moindre composant SIEM, une préparation rigoureuse de l'infrastructure évite la majorité des problèmes opérationnels. Cette phase est souvent négligée et cause plus d'incidents de production que les erreurs de configuration applicative.

Le **système d'exploitation de référence** pour ce guide est **Ubuntu 24.04 LTS (Noble Numbat)**, supporté jusqu'en avril 2029 (et 2034 avec ESM). Le choix d'Ubuntu 24.04 sur RHEL/Rocky 9 est guidé par la disponibilité des paquets officiels Wazuh et Graylog, la richesse de la documentation communautaire, et la compatibilité avec les dépôts OpenSearch AMI. Pour les environnements avec des exigences de conformité FIPS 140-2 ou des politiques Red Hat, Rocky Linux 9 est supporté par tous les composants avec des adaptations mineures.

La préparation commence par le **hardening OS** systématique avant installation des composants SIEM :

```
# Mise à jour complète du système
apt-get update && apt-get upgrade -y && apt-get dist-upgrade -y

# Désactivation des services inutiles
systemctl disable --now snapd avahi-daemon cups ModemManager
# ... (extrait — voir documentation officielle)
```

La **synchronisation temporelle** mérite une attention particulière : une désynchronisation de quelques secondes entre les sources de logs suffit à rendre les corrélations temporelles impossibles et à générer des faux positifs ou des faux négatifs dans les règles basées sur des fenêtres temporelles. Chrony est préféré à NTP classique pour sa convergence rapide après un décalage et sa résistance aux attaques de manipulation d'horloge.

4.2 Wazuh : installation All-in-One et migration vers l'architecture distribuée

L'approche recommandée est de démarrer avec le déploiement **All-in-One** pour valider la configuration, puis de migrer vers l'architecture distribuée si le volume d'agents le nécessite. Cette migration est

documentée et supportée par Wazuh, contrairement à certaines solutions qui piègent les utilisateurs dans leur déploiement initial.

L'installation All-in-One de Wazuh 4.10 utilise le script d'installation officiel qui configure Manager + Indexer + Dashboard sur un seul nœud :

```
# Téléchargement du script d'installation Wazuh 4.10 (vérifier la signature GPG)
curl -sO https://packages.wazuh.com/4.10/wazuh-install.sh
curl -sO https://packages.wazuh.com/4.10/wazuh-install.sh.sig
gpg --keyserver keyserver.ubuntu.com --recv-keys 0x8ACFE14B7A7B5BC3
gpg --verify wazuh-install.sh.sig wazuh-install.sh
# ... (extrait — voir documentation officielle)
```

Le script génère automatiquement une PKI interne (CA + certificats par composant), configure la communication TLS entre Manager, Indexer et Dashboard, et crée les index OpenSearch initiaux. Le Dashboard est accessible sur le port **443** (HTTPS) avec les credentials admin générés.

Le **hardening post-installation** de Wazuh est indispensable avant tout déploiement en production. Les points critiques :

```
# 1. Changement du mot de passe admin Wazuh Indexer
/usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-passwords-tool.sh \\
  -u admin -p 'VotreMotDePasseComplexe!2025'

# 2. Configuration de la rétention des index (ILM)
# ... (extrait — voir documentation officielle)
```

Le **déploiement des agents Wazuh** sur Windows se fait via GPO (déploiement MSI silencieux) ou SCCM/Intune pour les environnements géré. La commande d'enrôlement inclut l'adresse du Manager et optionnellement un groupe d'agents (pour l'application automatique de configurations par rôle machine) :

```
# Commande MSI Windows (déployée via GPO Software Installation)
# Wazuh-Agent-4.10.x-1.msi /q WAZUH_MANAGER="10.0.10.100" ^
#   WAZUH_MANAGER_PORT="1514" WAZUH_AGENT_GROUP="windows-servers" ^
#   WAZUH_REGISTRATION_SERVER="10.0.10.100" WAZUH_REGISTRATION_PORT="1515"

# ... (extrait — voir documentation officielle)
```

La configuration des **groupes d'agents Wazuh** est une fonctionnalité puissante souvent sous-utilisée. Un groupe est un ensemble de fichiers de configuration (ossec.conf partiel, règles supplémentaires, décodeurs personnalisés) appliqués automatiquement à tous les agents membres du groupe. Un groupe ”domain-controllers” configure l'audit renforcé des événements Kerberos et LDAP. Un groupe ”linux-servers” active la surveillance des logs SSH et des modifications de sudoers. Un groupe ”workstations” limite la verbosité pour éviter de noyer le SIEM dans du bruit Windows applicatif.

Points clés à retenir :

- Installation Wazuh 4.10 All-in-One via script officiel (vérifier la signature GPG) ; hardening post-install : nouveau mot de passe admin, ILM, audit logging OpenSearch.

- Groupes d'agents Wazuh : configurer ”domain-controllers”, ”linux-servers”, ”workstations” avec des politiques d'audit différenciées dès le départ.
- Synchronisation NTP chrony obligatoire sur tous les nœuds : une désynchronisation > 1s invalide les corrélations temporelles.
- Swap désactivé (swapoff -a + /etc/fstab) et vm.max_map_count=262144 sont des prérequis non-négociables pour OpenSearch.

4.3 Suricata 7.0 : installation depuis les sources OISF et configuration IDS/IPS

L'**Open Information Security Foundation (OISF)** maintient des dépôts officiels pour les principales distributions Linux. Sur Ubuntu 24.04 LTS, l'installation depuis le PPA OISF garantit d'avoir la dernière version stable de Suricata 7.x avec toutes les optimisations de performance :

```
# Ajout du PPA OISF officiel
add-apt-repository ppa:oisf/suricata-stable
apt-get update
apt-get install -y suricata suricata-update

# ... (extrait — voir documentation officielle)
```

La configuration du fichier **suricata.yaml** est l'étape la plus technique du déploiement Suricata. Les paramètres critiques pour un déploiement production :

```
# Extrait suricata.yaml — sections critiques à configurer

# 1. Définition des réseaux internes (HOME_NET)
vars:
  address-groups:
# ... (extrait — voir documentation officielle)
```

Pour le mode **IPS via NFQUEUE** (segments DMZ ou accès Internet), la configuration nftables dirige le trafic vers Suricata :

```
# Configuration nftables pour IPS NFQUEUE
# À adapter selon la topologie réseau

cat > /etc/nftables.d/suricata-ips.nft << 'NFTABLES'
table inet suricata {
# ... (extrait — voir documentation officielle)
```

La gestion des **faux positifs Suricata** est un défi opérationnel permanent. Les règles ET Open sont calibrées pour une large couverture, ce qui génère inévitablement des alertes sur du trafic légitime (notamment les règles SID 2000419 ICMP, SID 2013028 DNS, ou certaines règles SMB sur des environnements Windows intensifs). La méthode recommandée pour gérer les suppressions est le fichier `/etc/suricata/threshold.conf` avec des suppressions ciblées par SID + IP source/destination — jamais en désactivant la règle globalement :

```
# Exemple de suppressions ciblées (threshold.conf)
# Supprimer l'alerte SID 2013028 pour le serveur DNS interne connu
suppress gen_id 1, sig_id 2013028, track by_src, ip 10.0.10.53

# Limiter les alertes de scan interne à 1/minute par source
# ... (extrait — voir documentation officielle)
```

4.4 Graylog 6 : déploiement MongoDB Replica Set, OpenSearch cluster et tuning JVM

Le déploiement Graylog 6.x commence par ses dépendances : MongoDB pour les métadonnées et OpenSearch pour le stockage. L'ordre d'installation est important car Graylog vérifie la disponibilité de ses backends au démarrage.

Installation de **MongoDB 7.0** avec Replica Set (recommandé même en environnement test pour habituer les équipes aux procédures de production) :

```
# Import de la clé GPG MongoDB 7.0
curl -fsSL https://www.mongodb.org/static/pgp/server-7.0.asc | \\
  gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg ] \\
  https://repo.mongodb.org/apt/ubuntu noble/mongodb-org/7.0 multiverse" | \\
# ... (extrait — voir documentation officielle)
```

Installation d'**OpenSearch 2.x** pour le stockage des messages Graylog (distinct du Wazuh Indexer — ne pas partager le cluster OpenSearch entre Wazuh et Graylog en production) :

```
# Installation OpenSearch depuis le dépôt officiel
curl -o- https://artifacts.opensearch.org/publickeys/opensearch.pgp | \\
  gpg --dearmor | tee /usr/share/keyrings/opensearch-keyring.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/opensearch-keyring.gpg] \\
  https://artifacts.opensearch.org/releases/bundle/opensearch/2.x/apt stable main" | \\
# ... (extrait — voir documentation officielle)
```

Installation de **Graylog 6.1** sur Ubuntu 24.04 :

```
# Prérequis Java 17
apt-get install -y openjdk-17-jre-headless

# Dépôt Graylog officiel
wget -qO- https://downloads.graylog.org/repo/packages/graylog-6.1-repository_latest.deb \\
# ... (extrait — voir documentation officielle)
```

Le paramètre `-Dlog4j2.formatMsgNoLookups=true` dans la JVM Graylog est hérité de la mitigation **CVE-2021-44228 (Log4Shell)** — bien que les versions récentes de Graylog utilisent Log4j2 2.17+ qui n'est plus vulnérable, l'activer en défense en profondeur reste une bonne pratique.

Points clés à retenir :

- Graylog 6.1 requiert Java 17, MongoDB 7.0 (Replica Set 3 nœuds) et OpenSearch 2.x (cluster 3 nœuds) — les déployer dans cet ordre.

- Heap JVM Graylog : 8 Go pour 10 000 EPS ; heap OpenSearch : 50% de la RAM disponible, jamais > 31 Go. Ces deux règles sont non-négociables.
- message_journal_max_size=10gb : le journal Graylog absorbe les pics d'ingestion sans perte de logs — le dimensionner à 2x le volume d'ingestion horaire maximal.
- action.auto_create_index: false dans OpenSearch est obligatoire avec Graylog pour éviter la création d'index parasites qui consomment des shards.

4.5 Connexion Wazuh → Graylog via Filebeat GELF

L'intégration entre Wazuh et Graylog passe par **Filebeat**, le shipper léger d'Elastic qui lit les fichiers de logs du Manager Wazuh et les transmet à Graylog. Filebeat est installé sur le nœud Wazuh Manager — il lit les fichiers `/var/ossec/logs/alerts/alerts.json` (alertes Wazuh en JSON) et optionnellement `/var/ossec/logs/archives/archives.json` (tous les événements, verbeux).

```
# Installation Filebeat 8.x sur le serveur Wazuh Manager
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | \\
  gpg --dearmor | tee /usr/share/keyrings/elastic-keyring.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/elastic-keyring.gpg] \\
  https://artifacts.elastic.co/packages/8.x/apt stable main" | \\
# ... (extrait — voir documentation officielle)
```

Du côté **Graylog**, créer un Input de type **Beats** dans l'interface web (System → Inputs → Beats) en écoutant sur le port 5044 avec TLS activé. Une fois l'Input actif, les alertes Wazuh apparaissent dans les messages Graylog avec leurs champs JSON préservés.

Il faut ensuite créer un **Pipeline Graylog** pour normaliser les champs Wazuh vers l'ECS et router vers un Stream dédié :

```
# Règle de pipeline Graylog (créée via l'interface Web ou API)
# Extrait en Graylog Processing Language (GPL)

rule "Normalize Wazuh Alert to ECS"
when
# ... (extrait — voir documentation officielle)
```

4.6 Pipeline Suricata → Graylog avec parsing Eve JSON

L'envoi des événements Suricata vers Graylog utilise également Filebeat, avec le module Suricata intégré qui parse nativement le format Eve JSON :

```
# Configuration Filebeat sur le serveur Suricata
cat > /etc/filebeat/filebeat.yml << 'FILEBEAT_SURICATA'
filebeat.modules:
  - module: suricata
    eve:
# ... (extrait — voir documentation officielle)
```

Un pipeline Graylog dédié normalise les events Suricata. La particularité est que le champ `event_type` Eve JSON peut valoir ”alert”, ”dns”, ”http”, ”tls”, ”flow” — le pipeline route chaque type vers un stream approprié avec des politiques de rétention différentes (les flows réseau peuvent être conservés moins longtemps que les alertes IDS) :

```
# Règle GPL - Normalisation Suricata Alert
rule "Normalize Suricata IDS Alert"
when
  has_field("suricata.eve.event_type") AND
  to_string($message.suricata.eve.event_type) == "alert"
# ... (extrait — voir documentation officielle)
```

4.7 Automatisation du déploiement : Docker Compose, Ansible et mention Terraform

Bien que ce guide documente les installations manuelles pour en comprendre chaque étape, un déploiement en production reproductible doit être automatisé. Deux approches complémentaires sont recommandées :

Pour les environnements de **test et de validation**, un **Docker Compose** permet de démarrer l'ensemble du stack en quelques minutes :

```
# Structure du fichier docker-compose.yml pour le lab SIEM
# (version simplifiée — sans TLS pour le lab, OBLIGATOIRE en production)

version: '3.8'

# ... (extrait — voir documentation officielle)
```

Pour le déploiement en **production sur des serveurs bare-metal ou VMs**, les **Playbooks Ansible** garantissent la reproductibilité et facilitent la gestion des mises à jour. La structure recommandée organise les rôles Ansible par composant :

```
# Structure du projet Ansible
# siem-deployment/
# ├── inventory/
# │   ├── production.yml
# │   └── staging.yml
# ... (extrait — voir documentation officielle)
```

Pour les déploiements sur infrastructure cloud (**AWS, Azure, GCP**) ou dans des environnements où l'infrastructure elle-même doit être gérée comme du code, **Terraform** (ou OpenTofu, son fork open source) peut provisionner les VMs, réseaux, règles de pare-feu et volumes de stockage avant l'exécution des playbooks Ansible. L'approche GitOps — stocker le code Terraform et Ansible dans un dépôt Git, déclencher les déploiements via CI/CD — garantit la traçabilité et la reproductibilité des changements d'infrastructure, un prérequis pour les audits de conformité NIS2 et ISO 27001.

Recommandations complémentaires

Points clés à retenir :

- Filebeat est le shipper recommandé pour Wazuh → Graylog (alertes JSON) et Suricata → Graylog (Eve JSON) : il supporte le TLS mTLS, le retry automatique et le module Suricata natif.
- Les pipelines Graylog GPL normalisent vers l'ECS : c'est cette normalisation qui rend les corrélations cross-sources possibles dans les Event Definitions.
- Docker Compose pour les labs et la validation ; Ansible pour la production reproductible ; Terraform pour l'infrastructure-as-code cloud.
- Penser aux streams Graylog différenciés par type de source dès le départ : politiques de rétention, RBAC et règles d'alerte seront beaucoup plus simples à gérer.

À ce stade de la mise en œuvre, l'infrastructure SIEM hybride est opérationnelle dans sa configuration de base : Wazuh collecte les événements endpoint avec ses 3 200+ règles de détection, Suricata analyse le trafic réseau avec les 45 000+ signatures ET Open, et Graylog agrège, normalise et corrèle ces flux hétérogènes dans une interface SOC unifiée. Les alertes de niveau 9+ Wazuh et les alertes IDS Suricata de sévérité 1-2 remontent dans les streams Graylog dédiés, prêts à être enrichis et escaladés vers une plateforme de gestion d'incidents.

Cependant, cette configuration de base ne couvre pas encore les spécificités des environnements Active Directory hybrides : la collecte et l'analyse des événements Kerberos/NTLM sur les Domain Controllers, l'intégration des logs Microsoft 365 via Graph API, la détection des techniques d'attaque AD spécifiques (Pass-the-Hash, AS-REP Roasting, DCSync, Golden Ticket), et la corrélation cross-domain nécessitent une configuration avancée documentée dans la partie 2 de ce guide. Pour aller plus loin sur le déploiement Wazuh, consultez notre article dédié sur Wazuh SIEM/XDR open source : déploiement complet. La mise en conformité NIS2 de votre SIEM est abordée dans notre guide complet NIS2 et dans le guide ISO 27001. Pour les scénarios d'attaque Active Directory que ce SIEM devra détecter, référez-vous à notre section pentest Active Directory et aux procédures de réponse à incident.

Les ressources officielles indispensables pour approfondir chaque composant : documentation Wazuh 4.10, documentation Graylog 6.x, documentation Suricata 7.0 (OISF), le référentiel MITRE ATT&CK pour les TTPs à détecter, et les guides de l'ANSSI pour les recommandations de configuration souveraine.

5. Intégration Active Directory

L'intégration d'Active Directory dans un SIEM hybride constitue le cœur opérationnel de toute stratégie de détection des menaces internes. Active Directory demeure, en 2025, la cible prioritaire des groupes APT les plus sophistiqués : **Midnight Blizzard** (APT29, SVR russe), **Volt Typhoon** (APT41, MSS chinois) et **Storm-0539** (groupe de fraude financière) exploitent systématiquement les faiblesses de l'annuaire pour établir leur persistance, se déplacer latéralement et exfiltrer des données sensibles. La télémétrie AD représente donc la source de vérité absolue pour tout analyste SOC.

Point clé : Active Directory génère des centaines d'Event IDs distincts, mais seule une vingtaine concentre 90 % des signaux d'attaque pertinents. Maîtriser la sémantique de ces événements — et les corrélations entre eux — est plus important que de tout collecter.

5.1 Collecte des logs Windows

5.1.1 Wazuh Agent vs Winlogbeat vs Sysmon : recommandation de déploiement Le choix de l'agent de collecte sur les contrôleurs de domaine (DC) et les serveurs membres conditionne directement la richesse et la fiabilité de la télémétrie. Trois approches coexistent dans les architectures modernes, chacune avec ses compromis :

Wazuh Agent constitue la solution la plus intégrée pour un SIEM basé sur Wazuh. L'agent version 4.7+ collecte nativement les canaux Windows Event Log via le module winevt, supporte la détection de rootkits, l'inventaire système, la surveillance de l'intégrité des fichiers (FIM) et l'exécution de scripts de réponse active. Son principal avantage réside dans l'unification du plan de contrôle : un seul agent gère collecte, analyse locale et réponse. La limitation historique sur la richesse des champs extraits des événements Windows a été largement comblée depuis la version 4.5 avec l'introduction des *decoders* JSON natifs pour Sysmon.

Winlogbeat (Elastic) offre une collecte exhaustive et fiable des journaux Windows, avec un support natif de la stack ELK. Dans une architecture orientée Graylog, Winlogbeat peut transmettre directement via GELF ou Beats protocol. Cependant, cette approche introduit un agent supplémentaire à maintenir et ne bénéficie pas de l'écosystème de règles Wazuh ni des capacités de réponse active.

Sysmon (System Monitor, Microsoft Sysinternals) n'est pas un agent SIEM mais un pilote noyau qui génère une télémétrie réseau et processus extrêmement riche via le canal `Microsoft-Windows-Sysmon/Operational`. Sysmon est *complémentaire* à Wazuh Agent, pas alternatif. La configuration recommandée est celle de **Olaf Hartong** (sysmon-modular), une configuration modulaire maintenue activement qui couvre les techniques MITRE ATT&CK tout en limitant le bruit. Cette configuration génère typiquement les événements Sysmon 1 (création de processus), 3 (connexion réseau), 7 (chargement de DLL), 10 (accès à processus), 11 (création de fichier), 12/13/14 (registre), 22 (requête DNS) et 25 (falsification de processus, introduit dans Sysmon 15).

Table 5: Comparaison des solutions de collecte Windows pour environnements AD

Critère	Wazuh Agent seul	Wazuh + Sysmon (Hartong)
Richesse télémétrie processus	Moyenne (Event ID 4688 si audité)	**Très élevée** (Sysmon EID 1 + ligne de commande)
Réponse active intégrée	Oui (native Wazuh)	**Oui**
Overhead CPU/RAM sur DC	Faible (~1-2%)	Modéré (~3-5%)
Intégration règles Wazuh	Native	**Native (meilleure)**
Détection réseau (connexions)	Limitée	**Complète (Sysmon EID 3)**
Maintenance	Simple	Modérée (2 composants)
Coût	Gratuit	**Gratuit**

Recommandation opérationnelle : déployez Wazuh Agent 4.7+ couplé à Sysmon avec la configuration Hartong sur tous les contrôleurs de domaine et serveurs critiques. Sur les postes de travail, Wazuh Agent seul

suffit pour les environnements de taille moyenne, la configuration Hartong étant ajoutée progressivement sur les postes VIP ou exposés.

Bonnes pratiques

5.1.2 Canaux Windows critiques et politique d'audit avancée La politique d'audit avancée Windows (AAPC), configurable via GPO sous *Computer Configuration → Windows Settings → Security Settings → Advanced Audit Policy Configuration*, offre une granularité bien supérieure à l'audit classique. Sans cette configuration, de nombreux Event IDs critiques ne sont tout simplement pas générés.

Les catégories d'audit indispensables pour un SOC AD sont les suivantes. **Account Logon** : activez ”Audit Kerberos Authentication Service” (génère 4768, 4771) et ”Audit Kerberos Service Ticket Operations” (génère 4769, 4770) en succès ET échec. **Logon/Logoff** : ”Audit Logon” (4624, 4625, 4634, 4647) et ”Audit Special Logon” (4672) en succès et échec. **Account Management** : ”Audit User Account Management” (4720, 4722, 4723, 4724, 4725, 4726, 4738, 4740) et ”Audit Security Group Management” (4727-4735). **DS Access** : ”Audit Directory Service Access” (4662) et ”Audit Directory Service Changes” (5136, 5137, 5138, 5139, 5141) — critique pour DCSync et ACL abuse. **Object Access** : ”Audit File Share” (5140, 5145) pour détecter l'accès aux partages réseau. **Privilege Use** : ”Audit Sensitive Privilege Use” (4673, 4674).

Point clé : L'Event ID 4688 (création de processus) nécessite une activation spécifique de ”Audit Process Creation” ET l'activation de ”Include command line in process creation events” dans la GPO. Sans cette dernière option, l'Event ID 4688 est quasi-inutile pour la détection de commandes malveillantes.

Le canal **PowerShell** mérite une attention particulière. L'Event ID **4103** (Module Logging) capture les paramètres et sorties de chaque commande PowerShell exécutée. L'Event ID **4104** (Script Block Logging) capture le contenu complet des scripts, y compris après déobfuscation — ce qui en fait l'outil le plus puissant contre les techniques d'obfuscation PowerShell utilisées par les APT. Activez-les via GPO : *Computer Configuration → Administrative Templates → Windows Components → Windows PowerShell*. L'Event ID **4105** et **4106** (début/fin d'exécution de script) complètent le tableau. Notez que le Script Block Logging peut générer un volume important sur les serveurs Exchange ou SCCM — filtrez ces sources bruyantes dès la collecte.

Le canal **DNS Server** (`Microsoft-Windows-DNS-Server/Audit`) génère l'Event ID 541 pour chaque requête DNS résolue par le DC. Bien que volumineux, ce canal permet de détecter la reconnaissance DNS (Nmap, BloodHound), les connexions C2 via DNS over HTTPS (DoH) et l'exfiltration DNS. Configurez une rétention courte (24-48h) mais avec une indexation des domaines suspects.

Les canaux **Windows Firewall** (`Microsoft-Windows-Windows Firewall With Advanced Security/Firewall`) — Event IDs 2004, 2005, 2006 — permettent de détecter les modifications de règles pare-feu, technique couramment utilisée par les attaquants pour ouvrir des ports ou désactiver le pare-feu sur les cibles.

5.2 Détection des attaques classiques sur Active Directory

5.2.1 Attaques d'authentification **Kerberoasting (T1558.003)** est l'une des techniques les plus utilisées pour obtenir des hash de mots de passe de comptes de service sans interaction directe avec le contrôleur de domaine. L'attaquant, disposant d'un compte AD valide (même sans privilèges), demande un ticket de

service (TGS) pour tout compte ayant un SPN enregistré. Le ticket est chiffré avec le hash NTLM du compte de service, permettant un craquage hors ligne. Des groupes comme **Midnight Blizzard** ont documenté l'usage de Kerberoasting comme première étape d'élévation de privilèges (rapport CISA AA23-347A, décembre 2023).

La détection repose sur l'Event ID **4769** (Kerberos Service Ticket Operation) avec le champ `Ticket Encryption Type` valant **0x17** (RC4-HMAC). Les tickets légitimes modernes utilisent AES-256 (0x12) ou AES-128 (0x17 en 2003 compat). Une demande de TGS en RC4 depuis un compte non-service, vers un compte avec SPN, est un signal fort. Les outils comme Rubeus, Impacket GetUserSPNs.py et PowerView génèrent ce pattern de manière caractéristique.

AS-REP Roasting (T1558.004) cible les comptes avec le flag `DONT_REQ_PREAUTH` activé dans leurs propriétés UserAccountControl. Sans pré-authentification Kerberos, l'attaquant peut demander un AS-REP sans fournir de preuve d'identité — le DC répond avec un blob chiffré avec le hash du mot de passe du compte, craquable hors ligne. La détection via Event ID **4768** (TGT Request) sans Event ID 4771 correspondant (pré-auth échouée) est un indicateur. En pratique, cherchez des 4768 avec le champ `Pre-Authentication Type` à 0 (aucune pré-auth).

Brute-force et Password Spraying se distinguent par leur pattern temporel. Le brute-force classique génère des rafales d'Event ID **4625** (Logon Failure) suivi de l'Event ID **4740** (Account Locked Out) sur un même compte. Le password spraying, technique préférée des APT pour éviter les lockouts, génère un nombre limité de 4625 (1-3 tentatives) sur un grand nombre de comptes différents, sur une plage temporelle étendue. **Storm-0539** utilise systématiquement le password spraying contre les portails M365 mais aussi contre les contrôleurs de domaine exposés via VPN.

5.2.2 Mouvement latéral

Pass-the-Hash (T1550.002) permet à un attaquant d'utiliser un hash NTLM capturé pour s'authentifier sur d'autres machines sans connaître le mot de passe en clair. La détection exploite une nuance de l'Event ID **4624** : le Logon Type **9** (NewCredentials), combiné à un Authentication Package NTLM et un processus inhabituel (mimikatz, sekurlsa), indique typiquement un PtH. Le Logon Type **3** (Network) avec NTLM sur des comptes sensibles (administrateurs de domaine) depuis des machines inattendues est également suspect. L'Event ID **4648** (Logon with Explicit Credentials) capture les tentatives d'utilisation de credentials alternatifs.

Pass-the-Ticket (T1550.003) est plus difficile à détecter car les tickets Kerberos sont présentés de manière légitime. Les indicateurs incluent : un ticket TGT ou TGS présenté depuis une adresse IP différente de celle où il a été émis, un ticket avec une durée de vie anormalement longue (signe de forgeage), ou un ticket utilisé après expiration du compte. L'Event ID 4624 de Logon Type 3 avec Kerberos, depuis un compte normalement limité à des plages horaires ou des postes précis, est un signal d'alerte.

PSExec / SMB Lateral Movement (T1021.002) génère une combinaison caractéristique : Event ID **5145** (Network Share Object Access) sur `\\\\DC\\ADMIN$` ou `\\\\DC\\C$`, suivi de l'Event ID **7045** (Service Installed) dans le canal System du système cible. PSExec crée un service temporaire avec un nom aléatoire, l'exécute, puis le supprime — le tout en quelques secondes. Cette séquence est très fiable pour détecter les outils de type Impacket ou les variantes PSExec.

Points d'attention

Point clé : Pour le mouvement latéral via SMB, cherchez la séquence 5145 (accès partage ADMIN$) → 7045 (service installé) → 4624 Logon Type 3 → 7036 (service démarré) → 7045 (service supprimé) dans une fenêtre de 60 secondes. Ce pattern est quasi-pathognomonique de PSExec/Impacket.

5.2.3 Élévation de privilèges AD

DCSync (T1003.006) est une technique d'extraction des secrets Active Directory qui exploite le protocole de réplication MS-DRSR (Directory Replication Service Remote Protocol). L'attaquant simule le comportement d'un contrôleur de domaine pour demander la réplication des données d'un compte, incluant son hash NTLM. Cette technique est utilisée par Mimikatz (`lsadump::dcsync`), Impacket (`secretsdump.py`) et est documentée dans les TTP de **Midnight Blizzard** (NOBELIUM) lors de la compromission de SolarWinds.

La détection est précise : Event ID **4662** avec les propriétés d'accès incluant le GUID **1131f6aa-9c07-11d1-f79f-00c04fc2dcd2** (DS-Replication-Get-Changes) ou **1131f6ad-9c07-11d1-f79f-00c04fc2dcd2** (DS-Replication-Get-Changes-All), déclenché par un compte qui n'est pas un contrôleur de domaine. Le champ `Subject Account Name` doit correspondre à un compte non-DC pour que ce soit suspect. Activez l'audit sur l'objet de domaine lui-même via les SACL AD.

Golden Ticket (T1558.001) implique la compromission du hash du compte **KRBTGT** et la création de tickets TGT forgés valides pour n'importe quel compte. La durée de vie est typiquement très longue (10 ans dans les exemples Mimikatz). La détection est difficile car le ticket est techniquement valide. Les indicateurs incluent : un ticket TGT avec une durée de vie supérieure à la policy du domaine (généralement 10h), un SID utilisateur dans le ticket ne correspondant pas à l'annuaire, ou un Logon Type 3 depuis un compte désactivé ou inexistant. Microsoft a introduit la détection via les FAST (Flexible Authentication Secure Tunneling) et les Protected Users Security Group pour mitiger cette attaque.

DCShadow (T1207) est une technique avancée consistant à enregistrer un faux contrôleur de domaine pour injecter des modifications dans l'annuaire AD sans que celles-ci apparaissent dans les journaux classiques. La détection nécessite la surveillance des enregistrements SPN et des modifications des attributs `serverReference` dans la partition de Configuration AD, via Event ID 4742 (Computer Account Modified) sur des comptes non-DC.

5.2.4 Persistance dans Active Directory

ACL Abuse (T1222.001) est l'une des techniques de persistance les plus furtives. En modifiant les Access Control Entries (ACE) sur des objets AD sensibles (AdminSDHolder, compte krbtgt, GPO), un attaquant peut accorder des permissions abusives à un compte compromis qui survivront même à un changement de mot de passe. L'Event ID **5136** (Directory Service Object Modified) capture ces modifications, mais nécessite que l'audit DS Changes soit activé. Cherchez les modifications d'attributs `nTSecurityDescriptor` sur les objets sensibles (AdminSDHolder, Domain root, Domain Controllers OU).

Kerberos Delegation Abuse (T1558) : la délégation contrainte (Constrained Delegation, S4U2Proxy) et non contrainte (Unconstrained Delegation) sont des fonctionnalités légitimes souvent mal sécurisées. La compromission d'un serveur avec délégation non contrainte permet de capturer le TGT de tout utilisateur s'y connectant, y compris les administrateurs de domaine. Surveillez via Event ID 4624 sur les serveurs avec

délégation non contrainte, et via Event ID 4769 demandant des tickets pour des services sur des serveurs avec delegation.

5.3 Intégration Wazuh avec Active Directory

L'intégration de Wazuh avec Active Directory va au-delà de la simple collecte de logs. Wazuh supporte l'authentification LDAP/AD pour son interface web, permettant une gestion des accès basée sur les groupes AD. Configurez le fichier `/etc/wazuh-dashboard/opensearch_dashboards.yml` pour pointer vers vos contrôleurs de domaine LDAP, avec binding sécurisé LDAPS (port 636) plutôt que LDAP (389).

Le module syscollector de Wazuh construit un inventaire automatique des assets Windows : logiciels installés, ports ouverts, processus en cours, patches appliqués. Cet inventaire, stocké dans l'index `wazuh-states-inventory-*`, permet de contextualiser les alertes avec l'état de patch des machines concernées — crucial pour évaluer si une CVE active impacte un asset spécifique.

La synchronisation des groupes AD avec les agents Wazuh permet d'appliquer des politiques de configuration différenciées : les DC reçoivent une configuration d'audit maximale avec FIM sur SYSVOL et NTDS, tandis que les postes standard reçoivent une configuration allégée. Utilisez les groupes dynamiques Wazuh (`wazuh-agent-groups`) couplés à des scripts d'enrôlement qui lisent l'appartenance AD du machine account.

5.4 Règles de corrélation Wazuh spécifiques AD

5.4.1 Decoders personnalisés pour événements Windows AD Les decoders Wazuh transforment les logs bruts en champs structurés utilisables par les règles. Pour les événements Windows collectés via le module `winevt`, Wazuh utilise des decoders XML. Voici un decoder complet pour l'Event ID 4769 (Kerberos TGS Request) :

```
<!-- /var/ossec/etc/decoders/local_decoder.xml -->
<decoder name="windows-4769-kerberos-tgs">
  <parent>windows</parent>
  <type>windows</type>
  <prematch>Microsoft-Windows-Security-Auditing</prematch>
# ... (extrait — voir documentation officielle)
```

5.4.2 Règles personnalisées pour détections AD Les règles Wazuh s'appliquent aux événements décodés et génèrent des alertes avec un niveau de sévérité (0-15). Voici les règles essentielles pour la détection des attaques AD :

Architecture détaillée

```
<!-- /var/ossec/etc/rules/local_rules.xml -->
<!-- Groupe de règles pour Active Directory -->
<group name="windows,active_directory,">
```

```
  <!-- KERBEROASTING : TGS request avec chiffrement RC4 (0x17) -->
# ... (extrait — voir documentation officielle)
```

5.4.3 Conversion Sigma et intégration sigma-cli Les règles **Sigma** constituent un format standardisé de détection, indépendant du SIEM cible. Le dépôt SigmaHQ contient plus de 3000 règles maintenues par la communauté. La conversion vers le format Wazuh XML s'effectue via `sigma-cli` avec le backend `wazuh`.

```
# Règle Sigma pour Kerberoasting - sigma/rules/windows/builtin/security/kerberoasting.yml
title: Kerberoasting via RC4 Ticket Request
id: 56700ac3-8a79-4e5f-9d4c-bcbf04445bfd
status: stable
description: Detects Kerberoasting attack by monitoring TGS requests with RC4 encryption
# ... (extrait — voir documentation officielle)
```

La conversion s'effectue avec la commande suivante :

```
# Installation sigma-cli avec backend wazuh
pip install sigma-cli
pip install pySigma-backend-wazuh

# Conversion d'une règle unique
# ... (extrait — voir documentation officielle)
```

Point clé : Lors de la conversion Sigma → Wazuh, vérifiez systématiquement les règles générées. Le backend Wazuh peut produire des regex incompatibles ou des champs mal mappés. Testez chaque règle convertie avec `wazuh-logtest` avant déploiement en production.

5.5 Visualisation et alertes Graylog pour Active Directory

Graylog reçoit les alertes Wazuh via l'intégration décrite en partie 1, mais aussi directement les logs Windows bruts pour une corrélation indépendante. Les dashboards AD dans Graylog doivent couvrir quatre dimensions : l'authentification, les changements d'objets AD, les mouvements latéraux et les élévations de privilèges.

Pour le dashboard **Authentication Overview**, configurez des widgets visualisant : le ratio succès/échec des connexions par heure (courbe temporelle), les top 10 comptes avec échecs (tableau), la carte géographique des connexions par IP source, et les logons de type 9 (NewCredentials) isolés. Utilisez des streams dédiés avec un filtre sur `win_system_eventID:(4624 OR 4625 OR 4648 OR 4768 OR 4769)`.

Pour le dashboard **AD Changes Monitor**, les Event IDs 5136 (modification d'objet DS) et 4720 (création de compte) sont les plus critiques. Créez une alerte Graylog de type ”count” sur les 5136 avec `objectClass:user AND attributeLDAPDisplayName:userAccountControl` — toute modification du UserAccountControl (activation DONT_REQ_PREAUTH par exemple) doit déclencher une alerte immédiate.

5.6 Validation Atomic Red Team

Atomic Red Team de Red Canary propose des tests d'attaques atomiques mappés aux techniques MITRE ATT&CK. Ces tests permettent de valider que vos détections fonctionnent avant qu'un vrai attaquant ne les exploite. Voici les tests prioritaires pour valider les détections AD :

Table 6: Tests Atomic Red Team pour validation des détections AD

Technique	ID MITRE	Commande Atomic Red Team	Event IDs attendus
Kerberoasting	T1558.003	`Invoke-AtomicTest T1558.003 -TestNumbers 1`	4769 (0x17)
AS-REP Roasting	T1558.004	`Invoke-AtomicTest T1558.004 -TestNumbers 1`	4768 (PreAuth=0)
DCSync	T1003.006	`Invoke-AtomicTest T1003.006 -TestNumbers 1`	4662 (GUID réplicat
PowerShell encoded	T1059.001	`Invoke-AtomicTest T1059.001 -TestNumbers 2`	4104 (ScriptBlock)
Pass-the-Hash	T1550.002	`Invoke-AtomicTest T1550.002 -TestNumbers 1`	4624 (Type 9)
PSExec lateral	T1021.002	`Invoke-AtomicTest T1021.002 -TestNumbers 1`	7045, 5145

La procédure de validation recommandée est la suivante : exécutez chaque test Atomic Red Team dans un environnement AD de lab isolé, vérifiez que Wazuh génère l'alerte attendue avec le bon niveau de sévérité, puis comparez les faux positifs observés avec les exceptions à configurer. Documentez le taux de détection dans un registre de tests maintenu en Git.

Pour aller plus loin, consultez nos guides dédiés sur les attaques AD : Kerberoasting : attaque et défense, DCSync : anatomie et contre-mesures et Golden Ticket : détection et réponse.

6. Intégration Microsoft 365

L'intégration de Microsoft 365 dans le SIEM hybride représente un défi architectural spécifique : les logs M365 résident dans le cloud Microsoft, avec des API d'accès dédiées, des délais d'ingestion variables (jusqu'à 90 minutes pour l'UAL) et une sémantique d'événements radicalement différente des logs Windows on-premise. Pourtant, la corrélation entre les signaux AD on-premise et les activités M365 est essentielle pour détecter les attaques hybrides modernes, qui constituent le mode opératoire privilégié des groupes APT étatiques depuis 2023.

Point clé : Microsoft 365 E3 conserve les logs UAL 90 jours. E5 ou Audit Premium (add-on) étend cette rétention à 1 an et active les logs MailItemsAccessed (critique pour détecter la compromission de boîtes mail). Sans licence E5/Audit Premium, vous êtes aveugle sur les accès aux emails individuels.

6.1 Sources de logs Microsoft 365

6.1.1 Unified Audit Log (UAL) Le Journal d'audit unifié (UAL) centralise les événements de sécurité de l'ensemble des services M365 : Exchange Online, SharePoint, OneDrive, Teams, Azure AD, Intune, Defender.

Il doit être activé explicitement dans le Centre de conformité M365 — il n'est pas actif par défaut sur les anciens tenants. Les événements sont accessibles via l'API `Office 365 Management Activity API` et via PowerShell (`Search-UnifiedAuditLog`).

Les types d'opérations les plus critiques pour la sécurité incluent : **UserLoggedIn** et **UserLoginFailed** (authentifications Entra ID), **MailItemsAccessed** (accès aux emails, E5 uniquement), **Send** (emails envoyés), **FileAccessed** et **FileDownloaded** (SharePoint/OneDrive), **AddedToGroup** et **RemovedFromGroup** (modifications de groupes), **Consent to application** (consentements OAuth), **UpdateInboxRules** (règles de messagerie créées/modifiées), **eDiscovery search** (recherches eDiscovery potentiellement abusives).

6.1.2 Sign-in Logs Entra ID Les journaux de connexion Entra ID (anciennement Azure AD) fournissent des métadonnées de risque absentes de l'UAL classique. Chaque événement de connexion inclut : le **Risk Level** (Low/Medium/High) calculé par Microsoft Identity Protection, le **Risk Detail** (unfamiliarFeatures, anonymizedIPAddress, impossibleTravel, etc.), le résultat de l'évaluation des **Conditional Access Policies**, le type de client MFA utilisé, et la localisation géographique avec l'ASN (Autonomous System Number) permettant d'identifier les proxies VPN ou Tor.

Ces logs sont disponibles dans deux canaux distincts : les **Interactive sign-ins** (connexions utilisateur directes) et les **Non-interactive sign-ins** (connexions applicatives, refresh tokens, PRT). Ce second canal est souvent négligé mais critique pour détecter l'abus de Primary Refresh Tokens (PRT), technique documentée dans les TTP de **Midnight Blizzard**.

6.1.3 Audit logs spécifiques par service Chaque service M365 génère ses propres événements d'audit au sein de l'UAL. **Exchange Online** produit les opérations MailItemsAccessed, Send, UpdateInboxRules, AddMailboxPermission et MoveToDeletedItems. **SharePoint et OneDrive** génèrent FileAccessed, FileDownloaded, FileSyncUploadedFull, SharingSet (partage créé) et SharingInvitationCreated. **Microsoft Teams** produit des événements sur la création de canaux, l'ajout de membres, le partage de fichiers et les réunions enregistrées. Ces logs Teams sont particulièrement importants depuis que **MERCURY/MuddyWater** (APT iranien) a documenté l'utilisation d'équipes Teams pour des campagnes de social engineering (rapport Microsoft MSTIC, mai 2023).

6.2 Collecte via Wazuh

6.2.1 Module office365 natif Wazuh Wazuh 4.4+ inclut un module de collecte M365 natif qui interroge l'Office 365 Management Activity API. La configuration s'effectue dans `/var/ossec/etc/ossec.conf` sur le manager :

```
<!-- Configuration module office365 dans ossec.conf -->
<office365>
  <enabled>yes</enabled>
  <interval>5m</interval>
  <curl_max_size>10M</curl_max_size>
# ... (extrait — voir documentation officielle)
```

L'application Azure AD pour Wazuh doit disposer des permissions d'API suivantes (Application permissions, pas Delegated) : `ActivityFeed.Read`, `ActivityFeed.ReadDlp` et `ServiceHealth.Read` sur l'API *Office 365 Management APIs*. Ces permissions nécessitent un consentement administrateur du tenant. Notez que les droits sur Microsoft Graph sont distincts et nécessitent une configuration séparée pour les sign-in logs Entra ID.

6.2.2 Microsoft Graph API pour les sign-in logs Les journaux de connexion Entra ID ne sont pas accessibles via l'Office 365 Management API mais via **Microsoft Graph API**. Un script Python complémentaire collecte ces logs et les injecte dans Wazuh via le socket Unix local. Les permissions Graph requises sont : `AuditLog.Read.All` et `Directory.Read.All`. Consultez notre guide dédié : Automatiser l'audit de sécurité M365 avec PowerShell et Graph.

6.3 Suricata pour le trafic vers Microsoft

Suricata analyse le trafic réseau vers les services Microsoft cloud, permettant de détecter des comportements anormaux que les logs M365 ne capturent pas. Le chiffrement TLS ne masque pas le SNI (Server Name Indication) dans les versions antérieures à TLS 1.3 avec ECH (Encrypted Client Hello), et même en TLS 1.3, le certificate en clair dans le Server Hello révèle la destination.

Les signatures Suricata utiles pour M365 incluent la détection d'uploads massifs vers OneDrive (`api.onedrive.com`) en dehors des heures bureaux, les connexions vers des sous-domaines Microsoft inhabituels (ex. `*.blob.core.windows.net` depuis des postes non-admin), et l'utilisation de Microsoft Teams comme canal C2 — technique documentée par **MERCURY/MuddyWater** utilisant l'API Graph Teams pour l'exfiltration.

```
# Règle Suricata pour détection exfiltration OneDrive
# /etc/suricata/rules/microsoft365.rules

alert http $HOME_NET any -> $EXTERNAL_NET any (
    msg:"EXFIL Suspicious large upload to OneDrive/SharePoint";
# ... (extrait — voir documentation officielle)
```

Wazuh dispose de règles natives pour Suricata (fichier `0085-suricata_rules.xml`) qui traitent les alertes EVE et les mappent à des niveaux de sévérité Wazuh. Une alerte Suricata de sévérité 1 (critique) génère un event Wazuh de niveau 12, sévérité 2 → niveau 10, sévérité 3 → niveau 6. Ces mappings sont ajustables selon votre contexte.

L'intégration avancée exploite le champ **community-id** de Suricata — un hash standardisé (RFC draft) calculé sur le quintuplet réseau (proto, src IP, src port, dst IP, dst port) — pour corréler une alerte Suricata avec les logs applicatifs du même flux collectés par Wazuh sur l'endpoint destination. Cette corrélation réseau-endpoint est extrêmement puissante pour confirmer une exploitation.

7.3 Pipelines Graylog avancés

7.3.1 Extracteurs : regex, Grok et JSON Les extracteurs Graylog transforment les messages bruts en champs structurés, similairement aux decoders Wazuh. Trois types d'extracteurs sont disponibles. Les

extracteurs **JSON** parsent automatiquement les messages JSON valides en champs Graylog — idéaux pour les logs EVE Suricata ou les alertes Wazuh forwarded en JSON. Les extracteurs **Grok** utilisent des patterns nommés similaires à ceux de Logstash, avec une bibliothèque de patterns prédéfinis couvrant syslog, Apache, Nginx, Windows Event Log. Les extracteurs **Regex** permettent des extractions personnalisées via des groupes nommés.

Pour les logs Windows collectés par Wazuh et transmis à Graylog, configurez un extracteur JSON sur le champ `message`, puis des extracteurs de sous-champ pour normaliser les noms : `win.eventdata.targetUserName → user`, `win.eventdata.ipAddress → src_ip`, conformément au schéma ECS (Elastic Common Schema). Cette normalisation facilite la création de dashboards et règles d'alerte qui fonctionnent sur des sources hétérogènes.

7.3.2 Pipelines de traitement et lookup tables Les pipelines Graylog implémentent un langage de règles déclaratif permettant d'enrichir, filtrer, router et transformer les messages. Un pipeline se compose de stages (étapes) ordonnées, chacune contenant des règles conditionnelles. Voici un exemple de pipeline pour enrichir les événements AD avec les données d'asset :

```
// Pipeline Graylog : enrichissement AD
// Stage 1 : Identification des événements Windows Security
rule "identify_windows_security"
when
    has_field("win_system_channel") AND
# ... (extrait — voir documentation officielle)
```

Les **lookup tables** Graylog permettent d'enrichir les messages avec des données de référence externes : liste des assets (IP → hostname + owner + criticité), threat intelligence (IP → réputation + pays + ASN), liste des comptes de service (username → SPN + département). Ces tables sont chargées depuis des fichiers CSV, des bases de données (via adapter JDBC) ou des API REST. Mettez à jour les tables d'assets quotidiennement via un cron qui exporte depuis votre CMDB.

7.3.3 Streams, index sets et rétention Les streams Graylog routent les messages vers des index OpenSearch dédiés selon des critères de correspondance. Cette architecture permet d'appliquer des politiques de rétention différenciées selon la criticité des données. Recommandation de structuration :

Table 7: Architecture des streams et index sets Graylog recommandée

Stream	Critère de routage	Index set	Rétention chaude	Rétention fro
Security Alerts Critical	alert_severity:CRITICAL	security_critical	365 jours	5 ans (cold st
Windows Security Events	win_system_channel:Security	windows_security	90 jours	1 an
M365 Audit	source:office365	m365_audit	90 jours (E3) / 365 (E5)	10 ans (E5 Au
Suricata IDS	source:suricata	suricata_ids	30 jours	90 jours
Network Flows	event_type:flow	network_flows	14 jours	30 jours

Stream	Critère de routage	Index set	Rétention chaude	Rétention fro
Linux/Unix Systems	os_type:linux	linux_systems	60 jours	180 jours

7.3.4 Alerting multi-canal Graylog Graylog supporte plusieurs types d'alertes. Les alertes de **count** déclenchent lorsqu'un nombre d'événements dépasse un seuil sur une période. Les alertes d'**aggregation** permettent des conditions complexes (COUNT DISTINCT, SUM, AVG sur des champs). Les alertes **statistiques** détectent les anomalies par rapport à une baseline. Les notifications supportent Email, Slack, PagerDuty, Teams, et des webhooks HTTP personnalisés permettant d'appeler n'importe quel SOAR ou système tiers.

7.4 Framework MITRE ATT&CK : mapping et couverture

7.4.1 Tableau de couverture MITRE ATT&CK Mesurer la couverture MITRE ATT&CK de votre SIEM est un exercice indispensable pour identifier les angles morts. Voici le tableau de couverture de la stack Wazuh + Graylog + Suricata pour les tactiques Enterprise ATT&CK les plus pertinentes :

Table 8: Couverture MITRE ATT&CK Enterprise par composant SIEM

Tactique	Technique clé	ID	Wazuh	Suricata	Gra
Credential Access	Kerberoasting	T1558.003	Oui (4769)	Non	Co
Credential Access	DCSync	T1003.006	Oui (4662)	Partiel (SMB)	No
Lateral Movement	Pass-the-Hash	T1550.002	Oui (4624 T9)	Non	Co
Lateral Movement	SMB/PsExec	T1021.002	Oui (7045, 5145)	Oui (SID 2009030)	Co
Privilege Escalation	Golden Ticket	T1558.001	Partiel (4769)	Non	Co
Persistence	ACL Abuse	T1222.001	Oui (5136)	Non	En
Defense Evasion	Obfuscated Files	T1027	Oui (4104)	Partiel (HTTP)	Co
Exfiltration	Exfil over Web	T1567	Partiel (processus)	Oui (HTTP upload)	Co
Command and Control	Application Layer Protocol	T1071	Non	Oui (DNS, HTTP, TLS)	Co
Initial Access	Phishing (M365)	T1566.002	Partiel (consent grant)	Non	Co

7.4.2 Sigma : versioning Git et CI/CD pour règles Traiter les règles de détection comme du code — Detection as Code — est une pratique de maturité SOC essentielle. Vos règles Sigma, Wazuh XML et Suricata doivent être versionnées dans Git, avec des processus de revue, de test et de déploiement automatisés.

Structurez votre dépôt de règles ainsi : `sigma/` pour les règles Sigma sources, `wazuh/rules/` pour les XML compilés, `suricata/rules/` pour les règles .rules, `tests/` pour les logs de test et les scripts

de validation. Chaque règle doit avoir un identifiant unique, une date de création, une date de dernière modification et un champ `status` (test/stable/deprecated).

Un pipeline CI/CD GitLab ou GitHub Actions pour les règles de détection peut inclure : validation syntaxique des règles Sigma avec `sigma check`, conversion automatique vers les formats cibles (`sigma convert`), tests automatisés contre des logs de référence connus (attack simulations logs), déploiement progressif (staging → production) avec rollback automatique en cas d'augmentation anormale des faux positifs. Le dépôt SigmaHQ publie des règles pour les dernières CVE dans les 48-72h suivant leur découverte publique.

Point clé : Implémentez un workflow de gestion des faux positifs : chaque ticket "faux positif" doit aboutir soit à une modification de la règle déclenchante (si le signal est vraiment bénin), soit à l'ajout d'une exception documentée avec justification métier, soit à la création d'une règle de suppression temporelle. Ne supprimez jamais une règle entière pour éliminer des faux positifs — affinez-la.

7.5 YARA et renseignement sur les menaces

7.5.1 YARA dans Wazuh YARA est un outil de classification et d'identification de malwares via des règles de correspondance de patterns (chaînes, regex, conditions booléennes). Wazuh intègre YARA via le module `active-response` et le module `syscheck` : à chaque nouveau fichier créé sur un endpoint supervisé, un script YARA peut être déclenché pour analyser le fichier contre une base de règles.

Les sources de règles YARA recommandées incluent : Yara-Rules project (règles génériques), les règles de **Florian Roth** (Nextron Systems, maintenues activement pour les derniers malwares APT), les règles CAPE Sandbox et Any.run pour les familles de malwares récentes. En 2024-2025, des règles YARA ont été publiées pour détecter **SILKBEAN** (Android malware APT41), **CosmicDuke** (APT29) et les loaders **PIKABOT** utilisés par les affiliés de ransomware.

7.5.2 Threat intelligence : MISP et lookup tables Graylog L'intégration de flux de renseignement sur les menaces (threat intel) enrichit les alertes avec des indicateurs de compromission (IOC) connus. Wazuh supporte nativement l'intégration avec **MISP** (Malware Information Sharing Platform) et **VirusTotal**. Pour chaque nouveau fichier détecté par FIM ou chaque hash de processus capturé, Wazuh peut interroger MISP et VirusTotal pour vérifier la réputation.

Dans Graylog, les lookup tables alimentées par des flux OSINT permettent d'enrichir les events réseau en temps réel : chaque IP source est vérifiée contre Abuse.ch (Feodo Tracker, URLhaus), AlienVault OTX et les listes de Blocklist.de. Les IPs identifiées comme malveillantes reçoivent un champ `threat_intel_match:true` et une catégorie (botnet, ransomware-c2, phishing). Ces enrichissements transforment une alerte Suricata générique en alerte contextualisée avec attribution partielle.

7.6 Active Response et orchestration SOAR

7.6.1 Active Response Wazuh natif Le module active-response de Wazuh permet d'exécuter des scripts sur les agents en réaction à des alertes. Wazuh dispose de scripts natifs : `firewall-drop` (bloque une IP via iptables/nftables), `host-deny` (ajoute une entrée /etc/hosts.deny), `disable-account` (désactive

un compte local), `win_route-null` (route null une IP sur Windows). Ces actions sont configurées dans `ossec.conf` :

```
<!-- Active Response : blocage IP automatique sur alerte Suricata critique -->
<active-response>
  <command>firewall-drop</command>
  <location>all</location>
  <rules_id>100100,100101,100102</rules_id>
# ... (extrait — voir documentation officielle)
```

Les scripts d'isolation d'endpoint (Windows ou Linux) appliquent des règles iptables/Windows Firewall restrictives ne laissant passer que le trafic vers le SIEM et le contrôleur de domaine (pour permettre l'investigation). Cette isolation est réversible via une commande manager Wazuh, contrairement à une déconnexion physique du réseau.

7.6.2 TheHive + Cortex pour la gestion des cas **TheHive** est une plateforme open source de gestion des incidents de sécurité (case management). Elle s'intègre nativement avec Wazuh via un script Python qui crée automatiquement un case TheHive pour chaque alerte Wazuh de niveau 12+. Chaque case inclut les observables (IP, hashes, usernames), le contexte de l'alerte et un timeline des événements corrélés.

Cortex, le moteur d'analyse de TheHive, exécute des *analyzers* automatisés sur les observables : VirusTotal pour les hashes et URLs, Shodan pour les IPs, Abuse.ch pour les domaines, Have I Been Pwned pour les emails. Ces analyses enrichissent automatiquement le dossier de l'incident et accélèrent la triage. En 2025, Cortex propose plus de 140 analyzers communautaires couvrant l'ensemble des plateformes de threat intelligence majeures.

7.6.3 Webhooks SOAR : n8n et intégrations Python Pour les organisations sans budget SOAR commercial (Splunk SOAR, Palo Alto XSOAR), **n8n** (self-hosted, licence Elastic v2) ou **Node-RED** offrent des capacités d'orchestration viables. Un workflow n8n typique pour la gestion d'une alerte DCSync pourrait : recevoir le webhook Graylog → interroger l'API AD pour vérifier l'appartenance du compte aux groupes DA → créer un ticket TheHive → notifier sur Teams → déclencher une action Wazuh pour isoler l'agent → ouvrir un ticket Jira pour le remediation tracking.

Point clé : Toute action automatisée de réponse (blocage IP, isolation réseau, désactivation de compte) doit avoir un mécanisme de rollback testé. Un faux positif ayant isolé un serveur de production critique est aussi problématique qu'une vraie attaque non détectée. Implémentez des listes blanches immuables pour les assets critiques.

7.7 Machine Learning et détection comportementale

7.7.1 UEBA Wazuh Wazuh implémente des fonctionnalités basiques de UEBA via le mécanisme de *First Time Seen* (balise `<if_fts>` dans les règles) et les statistiques de fréquence. Une règle marquée FTS génère une alerte la première fois qu'un pattern est observé pour un agent ou un utilisateur donné, puis supprime les alertes pour les occurrences suivantes. Ce mécanisme permet de détecter les connexions vers

de nouveaux services, les nouveaux processus exécutés, ou les premières connexions depuis un nouveau pays.

Pour une UEBA plus avancée, le module `statistical` de Wazuh calcule des seuils adaptatifs sur les compteurs d'événements (fréquence de connexion, volume de fichiers accédés, taux d'erreur). Lorsque le volume observé dépasse la moyenne + N écarts-types, une alerte est générée. Ce système, bien que simple, est efficace pour détecter les scans internes ou les comportements anormaux de comptes de service.

7.7.2 OpenSearch Anomaly Detection L'index OpenSearch sous-jacent à Wazuh Dashboard supporte le plugin **Anomaly Detection**, qui implémente l'algorithme Random Cut Forest (RCF) pour la détection non supervisée d'anomalies dans les séries temporelles. Configurez des détecteurs d'anomalies sur : le volume d'alertes par heure (détecte les bursts d'activité nocturne), le nombre de connexions réseau par endpoint (détecte le beaconing C2), la taille moyenne des emails sortants (détecte l'exfiltration mail).

Le plugin fonctionne en deux phases : une phase d'entraînement (128 points par défaut) pour établir la baseline, puis une phase de détection en continu qui génère un *anomaly score* normalisé de 0 à 1. Configurez des alertes sur les scores supérieurs à 0.8 pour les features critiques. L'entraînement initial prend 24-48h selon la fréquence des données — prévoyez une période de calibration avant de mettre les alertes en production.

8. Visualisation, reporting et SOC opérationnel

Un SIEM techniquement excellent mais sans interface opérationnelle exploitable perd une grande partie de sa valeur. La visualisation n'est pas une couche cosmétique — c'est l'interface entre la machine et l'analyste humain. Des dashboards bien conçus réduisent le temps de triage, permettent une détection visuelle d'anomalies et soutiennent la communication avec la direction et les équipes métier. L'objectif est d'atteindre un MTTD (temps moyen de détection) inférieur à 4h pour les incidents critiques et un MTTR (temps moyen de réponse) inférieur à 24h pour les incidents majeurs.

8.1 Wazuh Dashboard vs Graylog : forces complémentaires

Wazuh Dashboard (basé sur OpenSearch Dashboards) et Graylog remplissent des rôles complémentaires dans l'architecture de visualisation. Il ne s'agit pas de choisir l'un ou l'autre, mais de les utiliser selon leurs forces respectives.

Table 9: Comparaison Wazuh Dashboard vs Graylog pour la visualisation SOC

Cas d'usage	Wazuh Dashboard	Graylog	Recomm
Alertes temps réel	Excellente (native)	Bonne (event definitions)	Wazuh p
Threat hunting ad hoc	Bonne (Lucene/KQL)	Excellente (recherche full-text, streams)	Graylog
Dashboards conformité	Excellente (PCI, GDPR, HIPAA natifs)	Moyenne (création manuelle)	Wazuh p

Cas d'usage	Wazuh Dashboard	Graylog	Recomm
Corrélation multi-sources	Moyenne (index Wazuh uniquement)	Excellente (toutes sources)	Graylog
Gestion des agents	Excellente (native)	Non applicable	Wazuh e
Reporting PDF automatisé	Bonne (reporting plugin)	Moyenne (export CSV/JSON)	Wazuh p
Investigation forensique	Bonne (FIM, SCA, inventory)	Excellente (pivoting, timeline)	Graylog

8.2 Dashboards croisés : Threat Landscape, Identity, Network, Compliance

Un SOC mature maintient quatre dashboards opérationnels permanents, accessibles en temps réel sur des écrans dédiés dans la salle d'opérations.

Le **Threat Landscape Dashboard** présente une vue globale de l'activité d'alerte : compteurs d'alertes par sévérité sur les dernières 24h (comparés à la veille), carte mondiale des connexions avec coloration selon le score de risque, timeline des alertes critiques, top 10 des règles les plus actives, et état de santé des composants du SIEM (uptime Wazuh agents, status Suricata, latence indexation Graylog).

Le **Identity & Access Dashboard** se concentre sur l'authentification et les comptes : taux de succès/échec des connexions AD (courbe horaire), top 10 des comptes avec le plus d'échecs, activité des comptes privilégiés (DA, EA, SA), connexions hors horaires bureaux, first-time-seen logons, et alertes Kerberos actives. C'est le dashboard le plus consulté dans un SOC orienté Active Directory.

Le **Network & Endpoint Dashboard** visualise les alertes Suricata par catégorie (exploitation, C2, scan, exfiltration), le top des sources et destinations d'alertes réseau, les connexions sortantes vers des IPs répertoriées dans la threat intel, et les processus suspects détectés sur les endpoints (via Sysmon).

Le **Compliance Dashboard** utilise les modules natifs Wazuh pour PCI DSS, HIPAA, GDPR et NIST CSF. Wazuh mappe automatiquement chaque alerte aux contrôles des référentiels pertinents. Ce dashboard est destiné aux RSSI et aux équipes d'audit, avec un affichage simplifié du taux de conformité par domaine et les items non conformes à traiter.

8.3 Threat Hunting : méthodologie et requêtes Lucene

Le threat hunting est une démarche proactive qui cherche des signes de compromission non détectés par les règles automatisées. Contrairement au monitoring passif, le hunting part d'une hypothèse (inspirée des TTPs connus des APT, des nouvelles CVE, ou des incidents récents dans le secteur) et la valide ou l'infirme par des requêtes ciblées.

La méthodologie recommandée en 4 étapes : **1) Hypothèse** — ex. "Des comptes de service utilisent peut-être RC4 pour les tickets Kerberos suite à une migration AD incomplète." **2) Requête** — construction de la requête Lucene dans Graylog ou Wazuh Dashboard. **3) Analyse** — examen des résultats, identification des vrais positifs vs comportements légitimes. **4) Documentation** — si une anomalie est confirmée, ouverture d'un incident TheHive ; sinon, création d'une règle d'exception documentée et potentiellement d'une règle de détection automatique pour les prochaines fois.

Exemples de requêtes Lucene pour threat hunting dans Wazuh Dashboard / Graylog :

```
// Hunting : Kerberoasting - comptes service avec RC4 dans les 7 derniers jours
win.system.eventID:4769 AND win.eventdata.ticketEncryptionType:"0x17"
AND NOT win.eventdata.serviceName:("krbtgt" OR "host" OR "$")
// Affiner : group by win.eventdata.serviceName, win.eventdata.ipAddress

# ... (extrait — voir documentation officielle)
```

Point clé : Le threat hunting doit être planifié, pas improvisé. Créez un calendrier de hunting avec au moins 2-3 sessions par semaine pour les équipes SOC de taille moyenne, chacune centrée sur une tactique MITRE ATT&CK spécifique. Documentez chaque session (hypothèse, requêtes, résultats) dans un registre Git pour capitaliser sur l'expérience accumulée.

8.4 Reporting automatisé

Wazuh Dashboard propose un module de reporting (plugin OpenSearch Reporting) permettant de planifier la génération automatique de rapports PDF ou CSV. Configurez des rapports hebdomadaires pour : le résumé des alertes critiques et leur statut de résolution, l'évolution des métriques de conformité (PCI, GDPR), le top 10 des menaces détectées, et l'état de patch des assets supervisés (via l'inventaire Wazuh).

Ces rapports PDF sont automatiquement envoyés par email aux parties prenantes définies — RSSI, DSI, DPO pour les rapports GDPR. Dans Graylog Enterprise, les *scheduled searches* permettent d'exporter des datasets CSV pour alimentation des tableaux de bord Power BI ou Tableau de la direction.

Un rapport mensuel SOC exhaustif doit couvrir : nombre total d'alertes par sévérité, taux de faux positifs (incidents invalidés / incidents ouverts), MTTD et MTTR moyens par catégorie d'incident, couverture MITRE ATT&CK (pourcentage de techniques avec règles actives), nouvelles règles déployées, règles retirées ou ajustées, et incidents notables du mois avec chronologie.

8.5 Intégration ticketing : Jira, ServiceNow et TheHive

L'intégration du SIEM avec les systèmes de ticketing ferme la boucle opérationnelle : chaque alerte validée devient un ticket traçable avec assignation, SLA, historique des actions et résolution documentée. Wazuh supporte l'intégration native avec Jira Cloud et ServiceNow via son module `integrations` dans `ossec.conf`.

Pour Jira, configurez le champ `priority` en fonction du niveau Wazuh (15 → P1 Critical, 12-14 → P2 High, 8-11 → P3 Medium) et mappez les groupes de règles aux composants Jira (active_directory → composant "Identity", suricata → composant "Network Security"). L'intégration bidirectionnelle — où la fermeture d'un ticket Jira met à jour le statut dans Wazuh — nécessite un développement custom via l'API Wazuh REST, mais représente un gain opérationnel significatif pour les équipes.

8.6 KPIs SOC : métriques opérationnelles clés

La mesure de l'efficacité du SOC repose sur un ensemble de KPIs standardisés. Ces métriques permettent d'évaluer la performance opérationnelle, de justifier les investissements et d'identifier les axes

d'amélioration prioritaires.

Table 10: KPIs SOC recommandés avec cibles pour un SOC hybride Wazuh/Graylog/Suricata

KPI	Définition	Formule	Cible initial
MTTD	Mean Time to Detect	Temps entre début incident et première alerte	< 8h
MTTR	Mean Time to Respond	Temps entre alerte et containment confirmé	< 48h
Taux FP	Faux positifs	Alertes invalidées / Alertes totales × 100	< 30%
Couverture MITRE	% techniques couvertes	Techniques avec règles actives / Techniques totales	20%
Backlog alertes	Alertes non traitées > 24h	Alertes niveau 10+ ouvertes > 24h	< 20
Taux couverture agents	Assets supervisés	Agents actifs / Assets totaux inventoriés	70%
Disponibilité SIEM	Uptime infrastructure	Heures disponibles / Heures totales	99%

Le **MTTD** est la métrique reine du SOC : elle mesure directement l'efficacité de la détection. Un MTTD élevé signifie que l'attaquant dispose de plus de temps pour se déplacer latéralement et exfiltrer des données. Les recherches de Mandiant (maintenant Google Cloud Security) montrent qu'en 2024, le dwell time médian global est de 10 jours — votre objectif SOC doit être de descendre sous ce seuil pour votre périmètre.

Le **taux de faux positifs** est aussi critique que le MTTD : un taux élevé génère de la fatigue d'alerte, conduit les analystes à ignorer des alertes et augmente mécaniquement le MTTR. Suivez ce KPI par règle individuelle, pas seulement globalement — une seule règle mal calibrée peut représenter 80% des faux positifs totaux.

Point clé : Calculez votre ROI SIEM en croisant le coût opérationnel du SOC avec la valeur des incidents détectés et contenus. Un incident de ransomware contenu en 4h (grâce au MTTD de 2h) qui aurait coûté 2M€ sans détection justifie à lui seul plusieurs années d'investissement dans l'infrastructure SIEM.

Le **backlog d'alertes** est un indicateur de capacité opérationnelle : si les analystes ne peuvent pas traiter les alertes dans les SLA définis, soit les effectifs sont insuffisants, soit le taux de faux positifs est trop élevé, soit les priorités sont mal calibrées. Un backlog croissant est un signal d'alarme nécessitant une action corrective immédiate — automatisation, recrutement ou tuning des règles.

Pour le pilotage mensuel, présentez ces KPIs sous forme de tableau de bord direction avec une évolution sur 12 mois rolling. Segmentez par domaine (Identity/AD, M365, Network, Endpoints) pour identifier les axes d'amélioration prioritaires. Comparez avec les benchmarks sectoriels publiés par le MITRE ATT&CK Evaluations et les rapports annuels de Mandiant, CrowdStrike et IBM X-Force pour contextualiser votre performance relative.

Point clé : Un SIEM hybride open source bien opéré — Wazuh + Graylog + Suricata — peut atteindre 80-90% des capacités de détection d'une solution commerciale à un coût d'infrastructure 10 à 20 fois inférieur. L'investissement principal est humain : la qualité des règles, la rigueur du tuning et la compétence des analystes SOC font la différence, pas le coût de la licence.

9. Bonnes pratiques, hardening et conformité

Déployer une stack SIEM hybride open source représente un investissement technique considérable. La pérennité de cette infrastructure repose sur trois piliers indissociables : le **hardening** rigoureux de chaque composant, une gestion proactive des **faux positifs** et l'alignement continu avec les **référentiels réglementaires** applicables. Cette section aborde ces dimensions avec le niveau de profondeur qu'exige un déploiement en environnement de production.

9.1 Hardening des composants

Le hardening n'est pas une opération ponctuelle effectuée lors du déploiement initial : c'est un processus continu, documenté et vérifiable. La maxime des équipes sécurité expérimentées s'applique pleinement ici — *« un outil de sécurité non sécurisé devient lui-même un vecteur d'attaque »*. L'histoire récente donne raison à cette prudence : des instances Elasticsearch exposées sans authentification ont conduit à des fuites massives de logs, révélant paradoxalement les données que l'on cherchait à protéger.

9.1.1 Hardening OS — Ubuntu 24.04 LTS Ubuntu 24.04 LTS constitue le socle recommandé pour cette stack en 2025-2026. Sa politique de support étendu jusqu'en 2029 (ESM jusqu'en 2034) garantit la continuité des correctifs de sécurité sur la durée de vie d'un projet SOC typique.

Le référentiel **CIS Benchmarks for Ubuntu Linux 24.04 LTS** (niveau 2, profil serveur) fournit 250+ contrôles organisés en catégories : partitionnement sécurisé, services inutiles désactivés, durcissement réseau, gestion des comptes, auditing. L'outil Lynis — scanner de sécurité open source — permet d'évaluer rapidement le niveau de conformité via un score de 0 à 100 :

```
# Audit Lynis complet avec rapport détaillé
lynis audit system --cronjob --quiet --logfile /var/log/lynis.log
lynis audit system --check-all --report-file /var/log/lynis-report.dat

# Extraction des points critiques
grep "WARNING\\|SUGGESTION" /var/log/lynis.log | head -50
```

Les contrôles prioritaires pour les serveurs hébergeant Wazuh, Graylog ou Suricata sont :

- **Partitionnement** : /tmp, /var, /var/log, /var/log/audit sur partitions séparées avec options noexec, nosuid, nodev pour /tmp. Les bases de données Graylog (OpenSearch/MongoDB) sur volumes dédiés avec quotas stricts.
- **Kernel hardening** : sysctl.conf avec net.ipv4.conf.all.rp_filter=1, kernel.randomize_va_space=2, fs.suid_dumpable=0, kernel.kptr_restrict=2. AppArmor activé et profils configurés pour chaque service.
- **SSH** : authentification par clé uniquement, PermitRootLogin no, AllowUsers liste restrictive, Port non standard, bannière légale, MaxAuthTries 3, ClientAliveInterval 300.
- **Mises à jour automatiques** : unattended-upgrades configuré pour les correctifs de sécurité uniquement, avec redémarrage automatique les week-ends hors fenêtres de production. Les mises à jour majeures restent manuelles et testées sur environnement de pré-production.

OpenSCAP (Security Content Automation Protocol) complète Lynis en permettant une évaluation automatisée par rapport à des profils standardisés XCCDF. Son intégration dans les pipelines CI/CD garantit que toute nouvelle image serveur respecte la baseline sécurité avant mise en production.

Mise en pratique

Point clé — Hardening OS : Un score Lynis inférieur à 70/100 sur un serveur SIEM est inacceptable. Visez 80+ en production. Automatisez les audits hebdomadaires et intégrez les alertes de régression dans Graylog lui-même — surveiller le SIEM avec le SIEM est une bonne pratique de résilience.

9.1.2 Hardening applicatif **Wazuh — RBAC et isolation :** Wazuh intègre depuis la version 4.3 un système de contrôle d'accès basé sur les rôles (RBAC) complet. En production, créez des rôles fonctionnels distincts : analyste SOC N1 (lecture seule alertes), analyste N2 (gestion règles), ingénieur SIEM (administration complète), auditeur (accès logs bruts en lecture). L'API Wazuh doit être exposée uniquement en interne, avec certificats TLS mutuels (mTLS) entre le manager et les indexers. Les credentials par défaut (admin/admin) doivent être changés immédiatement à l'installation — une évidence souvent négligée dans les déploiements rapides.

Graylog — Tokens API et session management : Remplacez l'authentification par mot de passe par des **tokens API** pour toutes les intégrations programmatiques. Graylog 5.x supporte l'authentification LDAP/Active Directory et SAML — activez l'un ou l'autre pour centraliser la gestion des identités. La session expiration doit être configurée à 8 heures maximum. Activez le chiffrement TLS pour les API REST et GELF. MongoDB, base de données interne de Graylog, doit impérativement être configurée avec authentification (--auth) et accès réseau limité à localhost uniquement.

Suricata — Sonde isolée : L'interface de capture Suricata doit être configurée en mode promiscuité sur une interface réseau dédiée, distincte de l'interface de management. Cette séparation physique ou logique (VLAN dédié) empêche un attaquant ayant compromis la sonde réseau d'accéder directement au réseau de management du SIEM. Suricata doit fonctionner sous un compte système dédié sans shell (nologin), avec répertoires de logs en permission 750 et appartenant audit groupe suricata uniquement.

Point clé — MongoDB : CVE-2024-1234 (hypothétique illustration) et plusieurs vulnérabilités réelles documentées en 2024-2025 visent MongoDB sans authentification. Une instance MongoDB Graylog exposée sur 0.0.0.0 sans auth représente une exfiltration garantie de tous vos logs de sécurité. Vérifiez avec : `netstat -tlnp | grep 27017` — si vous voyez 0.0.0.0:27017, corrigez immédiatement.

9.1.3 Hardening réseau L'architecture réseau du SIEM doit suivre le principe de segmentation stricte. Un **VLAN dédié SIEM** (ex. VLAN 200, 10.200.0.0/24) héberge l'ensemble des composants. Les flux sont contrôlés par des ACL ou un pare-feu interne :

Table 11: Matrice de flux réseau SIEM — ports autorisés

Source	Destination	Port/Proto	Justification
Agents Wazuh (tous VLANs)	Wazuh Manager	1514/TCP, 1515/TCP	Events + enrollment
Switches/Routeurs	Graylog	514/UDP, 514/TCP (Syslog)	Logs équipements réseau
Applications	Graylog	12201/UDP (GELF)	Logs applicatifs structurés
Suricata	Graylog	5044/TCP (Beats)	Alertes IDS
SOC Analysts	Graylog UI / Wazuh UI	443/TCP	Interface web (via reverse proxy)
VLAN SIEM interne	Internet (OSINT)	443/TCP sortant	Threat intel feeds uniquement, via p
ALL → VLAN SIEM	VLAN SIEM	DENY par défaut	Isolation stricte

Le concept de **monitoring du monitoring** mérite une attention particulière. Si votre SIEM tombe en panne, qui vous alerte ? Configurez un heartbeat externe indépendant (Uptime Kuma, Nagios sur un serveur distinct) qui vérifie périodiquement que Graylog ingère toujours des événements. Un silence soudain des logs peut indiquer soit une attaque en cours qui désactive le SIEM, soit une panne silencieuse — les deux scénarios sont critiques.

9.2 Gestion des faux positifs

La gestion des faux positifs est souvent citée comme la première cause de **fatigue des alertes** (alert fatigue) dans les équipes SOC. Une étude IBM Security de 2024 révèle que les analystes SOC passent en moyenne 32% de leur temps sur des faux positifs. Cette inefficacité a un coût direct : 67% des analystes SOC junior quittent leur poste dans les 18 premiers mois, principalement en raison de la charge cognitive liée aux alertes non pertinentes.

9.2.1 Stratégie en 3 phases

Phase 1 — Observation (semaines 1-4) : Activez toutes les règles en mode « génération d'alertes uniquement » sans action automatique. Mesurez le volume d'alertes par catégorie, par source, par heure de la journée. Identifiez les « top 10 faux positifs » qui représentent généralement 80% du bruit (loi de Pareto appliquée à la cybersécurité). Ne touchez rien pendant cette phase — vous construisez une baseline.

Phase 2 — Ajustement (semaines 5-12) : Traitez les faux positifs par priorité décroissante de volume. Pour chaque ajustement, documentez : la règle modifiée, la justification technique, la date, l'auteur et la révision planifiée. Utilisez des whitelists ciblées plutôt que de désactiver des règles entières. Dans Wazuh, les balises <list> permettent des listes d'exceptions dynamiques (adresses IP, noms d'utilisateurs, hashes de processus) sans modifier les règles elles-mêmes.

Phase 3 — Révision périodique (mensuelle puis trimestrielle) : Les whitelists et suppressions doivent expirer. Une exception ajoutée pour un outil de scan de vulnérabilités lancé une fois peut masquer une

attaque réelle six mois plus tard. Planifiez des revues calendaires avec un propriétaire désigné pour chaque exception.

Point clé — Faux positifs : La corrélation avec la CMDB (base de données de gestion des configurations) est l'outil le plus puissant pour réduire le bruit. Une alerte de connexion administrative sur un serveur listé en CMDB comme « serveur d'administration » est moins critique qu'une connexion identique sur un serveur de production applicatif. Enrichissez systématiquement vos alertes avec des métadonnées CMDB.

9.3 Conformité réglementaire

La stack Wazuh + Graylog + Suricata couvre naturellement plusieurs exigences réglementaires, mais la conformité ne se décrète pas — elle se démontre par des preuves documentées et des processus vérifiables.

9.3.1 RGPD — Protection des données personnelles Les logs de sécurité contiennent par définition des données à caractère personnel (DCP) : adresses IP, noms d'utilisateurs, horodatages d'activité, contenus de sessions. Le **RGPD** impose plusieurs contraintes spécifiques aux SIEM :

- **Base légale** : L'intérêt légitime (Art. 6.1.f) constitue généralement la base légale pour la journalisation à des fins de sécurité. Le registre des traitements doit mentionner explicitement ce traitement avec sa durée de conservation et ses destinataires.
- **Durée de conservation** : Définissez des politiques de rétention strictes. Les logs d'accès : 6 mois recommandés (12 mois maximum pour certains secteurs réglementés). Les alertes de sécurité : 12-24 mois. Les logs d'investigation post-incident : durée de la procédure judiciaire éventuelle + 2 ans.
- **Droit à l'effacement** : Techniquement complexe avec OpenSearch/Elasticsearch. Implémentez une **pseudonymisation** des identifiants utilisateurs plutôt qu'un effacement au fil de l'eau — conservez un mapping chiffré identifiant pseudonyme ↔ identité réelle accessible uniquement sur réquisition.
- **Transferts hors UE** : Si vous utilisez des feeds de threat intelligence hébergés aux États-Unis, vérifiez les clauses contractuelles standard (SCC) ou privilégiez des sources européennes (ANSSI, CERT-FR, ENISA).

9.3.2 ISO 27001:2022 — Contrôles pertinents

Table 12: Mapping ISO 27001:2022 — Contrôles couverts par la stack SIEM

Contrôle ISO 27001:2022	Titre	Couverture par la stack	Preuves générées
A.5.7	Threat intelligence	Complète	Feeds MISP intégrés, IOC match
A.8.15	Logging	Complète	Centralisation Graylog, politiqu
A.8.16	Monitoring activities	Complète	Dashboards temps réel, alertes
A.5.24	Planification réponse incident	Partielle	Runbooks SOC (Annexe C), inté
A.5.25	Évaluation et décision incidents	Complète	Workflow triage Graylog, classif

Contrôle ISO 27001:2022	Titre	Couverture par la stack	Preuves générées
A.5.26	Réponse aux incidents	Complète	Active Response Wazuh, isolatic
A.5.27	Retour d'expérience incidents	Partielle	Post-mortems, métriques MTTD
A.8.7	Protection contre malwares	Complète	YARA Wazuh, Suricata signature
A.8.23	Web filtering	Partielle	Suricata HTTP inspection

Pour une certification ISO 27001, la stack SIEM doit être accompagnée d'une **politique de gestion des logs** formelle, d'un **registre des incidents** (même mineurs), et de preuves de tests réguliers des procédures de réponse. Consultez notre guide complet ISO 27001 : Guide Complet de Certification pour le périmètre organisationnel.

Optimisations avancées

9.3.3 NIS2 — Directive européenne de sécurité des réseaux La directive **NIS2** (transposée en droit français par ordonnance en 2024) impose aux entités essentielles (EE) et entités importantes (EI) des obligations de journalisation et de notification aux contours précis. Notre article Conformité NIS2 : Guide Pratique détaille le périmètre des entités concernées.

Points critiques NIS2 pour votre SIEM :

- **Notification 24h/72h** : Les incidents significatifs doivent être notifiés à l'ANSSI (ou CERT-FR selon l'entité) dans les 24 heures pour un premier rapport, et 72 heures pour un rapport complet. Votre SIEM doit générer automatiquement un rapport d'incident structuré (template NIS2) dès qu'une alerte de criticité élevée est confirmée.
- **Journalisation obligatoire** : NIS2 exige une journalisation des accès aux systèmes critiques, des modifications de configuration, et des événements de sécurité. Graylog + Wazuh couvrent ces exigences nativement.
- **Mesures de gestion des risques** : Le SIEM doit s'inscrire dans un processus formel d'évaluation des risques documenté et révisé annuellement.

9.3.4 LPM/SIIV/OIV — Exigences ANSSI Les **Opérateurs d'Importance Vitale** (OIV) et les **Systèmes d'Information d'Importance Vitale** (SIIV) sont soumis à la Loi de Programmation Militaire et aux arrêtés sectoriels ANSSI. Ces exigences vont au-delà de NIS2 :

- Déclaration obligatoire des incidents à l'ANSSI (pas seulement au CERT-FR)
- Journalisation des événements avec horodatage synchronisé NTP (source de temps de confiance)
- Conservation des logs 12 mois minimum sur support immuable (WORM)
- Interdiction de sous-traiter les logs de sécurité hors Union Européenne sans autorisation
- Audit de sécurité de la stack SIEM elle-même (prestataires PASSI qualifiés ANSSI)

Point clé — OIV/SIIV : Si votre organisation est qualifiée OIV ou gère des SIIV, l'utilisation de composants open source pour le SIEM est acceptable à condition de démontrer la maîtrise de la chaîne de dépendances

(SCA — Software Composition Analysis) et la capacité à appliquer des correctifs en moins de 72h pour les CVE critiques. Maintenez un inventaire SBOM (Software Bill of Materials) de votre stack.

9.4 Tests d'intrusion et validation

Un SIEM non testé est un SIEM dont on ignore s'il fonctionne. Les exercices de validation doivent être planifiés et documentés, avec des scénarios couvrant les **techniques MITRE ATT&CK** les plus fréquemment utilisées par les groupes APT ciblant votre secteur.

Les **exercices Red Team** testent la détection end-to-end : un consultant externe simule un attaquant réel, sans connaissance préalable de l'environnement, et tente d'atteindre des objectifs définis (accès aux données critiques, persistence, exfiltration). Le SIEM doit détecter et alerter avant que l'objectif soit atteint. Si ce n'est pas le cas, les règles et la visibilité doivent être renforcées.

Les **exercices Purple Team** sont plus collaboratifs : l'équipe offensive (Red) et l'équipe défensive (Blue) travaillent ensemble, le Red exécute une technique, le Blue vérifie en temps réel si elle est détectée. Ce format permet d'itérer rapidement sur les règles de détection et d'améliorer la couverture ATT&CK de manière méthodique.

Les **table top exercises** (exercices sur table) ne nécessitent aucun outillage technique : l'équipe SOC se réunit autour d'un scénario d'incident fictif et dérouile les procédures de réponse verbalement. Ces exercices révèlent les lacunes organisationnelles (qui appelle qui ? qui a les droits d'isolation ?) indépendamment des lacunes techniques. Recommandés trimestriellement pour les équipes SOC.

Point clé — Tests de détection : Atomic Red Team (projet open source de Red Canary) fournit des techniques d'attaque atomiques exécutables en une commande, mappées sur MITRE ATT&CK. Intégrez une batterie de tests Atomic Red Team dans vos pipelines CI/CD de la configuration SIEM — si une mise à jour de règle casse une détection existante, vous le saurez avant la mise en production.

9.5 Coûts et dimensionnement TCO

L'argument économique est souvent le premier avancé pour justifier une stack open source. Il mérite une analyse honnête et nuancée, car le coût réel d'un SIEM open source est loin d'être nul.

Coûts licences — La comparaison évidente :

Table 13: Comparatif licences SIEM — Solutions propriétaires vs open source (2025)

Solution	Modèle de tarification	Coût indicatif	Notes
Splunk Enterprise	Par volume ingéré (GB/jour)	150-300€/GB/mois	EPS illimité, storage costly
Microsoft Sentinel	Par GB ingéré	2-5€/GB ingestion + storage	Gratuit si déjà dans M365 E
IBM QRadar	Licences EPS (events/sec)	100-500k€/an enterprise	Très complexe, services pro
Elastic SIEM (ESS)	Par nœud ou GB stocké	1000-3000€/mois cluster	Open source mais cloud = p

Solution	Modèle de tarification	Coût indicatif	Notes
Exabeam	Par utilisateur/mois (UEBA)	20-50€/user/mois	Spécialisé UEBA, SOAR incl
Wazuh + Graylog + Suricata	**Licences**	**0€**	**Coûts infra et exploitation**

TCO réaliste sur 3 ans : Les vrais coûts d'une stack open source se cachent dans l'infrastructure, l'exploitation et la formation. Le tableau suivant présente une estimation réaliste pour trois tailles d'organisation :

Table 14: TCO estimatif sur 3 ans — Stack Wazuh + Graylog + Suricata

Poste de coût	100 endpoints	500 endpoints	2000 endpoints
Serveurs (achat ou cloud 3 ans)	8 000 €	25 000 €	90 000 €
Stockage (6 mois de logs)	2 000 €	8 000 €	35 000 €
Réseau (débit, infra)	500 €	2 000 €	8 000 €
Exploitation SIEM (ETP/an × 3)	60 000 € (0,4 ETP)	120 000 € (0,8 ETP)	360 000 € (2,4 ETP)
Formation initiale	3 000 €	6 000 €	15 000 €
Support communautaire/pro	0-5 000 €	5 000-20 000 €	20 000-60 000 €
Total 3 ans (fourchette basse)	**73 500 €**	**166 000 €**	**528 000 €**
Équivalent Splunk 3 ans (estimation)	150 000 €	500 000 €	2 500 000 €
Équivalent Sentinel 3 ans	40 000 €	120 000 €	450 000 €

Point clé — TCO honnête : Pour 100 endpoints, Microsoft Sentinel (intégré M365 E5) peut être moins cher qu'une stack open source si vous comptez le temps d'exploitation. L'avantage économique de l'open source devient décisif à partir de 500 endpoints, et massif au-delà de 2000. Pour les petites structures, le vrai argument n'est pas le coût mais la **souveraineté** et la **personnalisation**.

10. Études de cas détaillées

Les quatre études de cas suivantes sont composites, construites à partir de scénarios réels documentés dans la littérature de réponse à incident de 2023-2025. Les noms, secteurs et détails spécifiques ont été modifiés. Ces cas illustrent la valeur opérationnelle de la stack hybride dans des contextes d'attaque contemporains.

10.1 Cas 1 — Compromission Entra ID avec mouvement latéral AD

Contexte Une entreprise de services financiers de 1500 employés, présente en France et en Belgique. Infrastructure hybride : Active Directory on-premise (Windows Server 2022, 12 contrôleurs de domaine)

synchronisé avec Microsoft Entra ID (Azure AD) via **AD Connect** en mode Password Hash Synchronization (PHS). Microsoft 365 E3 pour la messagerie et la collaboration. La stack SIEM (Wazuh + Graylog + Suricata) est opérationnelle depuis 8 mois avec une équipe SOC de 3 analystes (2 N1, 1 N2).

Le groupe APT responsable présente des TTPs cohérents avec **Storm-0558** (groupe chinois documenté par Microsoft en 2023) combinées à des techniques de phishing ciblé attribuées à des acteurs à motivation financière. L'analyse post-incident suggère une compromission de type *Business Email Compromise* (BEC) évoluant vers une persistence avancée.

Vecteur initial Un email de phishing ciblé (spear phishing) est envoyé le lundi matin à 08h47 à un administrateur cloud de niveau 2. L'email imite parfaitement une notification Microsoft Authenticator concernant un appareil non reconnu — graphisme, domaine expéditeur usurpé via typosquatting (microsofft-security[.]com), certificat TLS valide. Le lien redirige vers une page de phishing adversary-in-the-middle (AiTM) utilisant Evilginx3 pour capturer le cookie de session post-MFA.

Chronologie détaillée **T+0 (Lundi 08h47)** — L'administrateur clique sur le lien de phishing. Evilginx3 capture les credentials et le cookie de session MFA. L'attaquant dispose d'un accès authentifié complet à Microsoft 365 et Entra ID de la victime. Aucune alerte générée à ce stade — la connexion semble légitime depuis Microsoft Graph.

T+15min (09h02) — L'attaquant interroge Graph API pour énumérer les groupes Entra ID, les membres du groupe « Global Administrators », et les applications enregistrées avec des permissions élevées. Volume de requêtes Graph anormal (142 appels en 8 minutes vs moyenne de 12/jour pour ce compte). **Première alerte Graylog** : règle « Microsoft 365 — Graph API enumeration burst » — criticité MEDIUM. L'analyste N1 de permanence note l'alerte mais la classe en observation — comportement « inhabituel mais pas impossible » pour un administrateur.

T+45min (09h32) — L'attaquant crée une nouvelle application Entra ID (nom : « Microsoft Teams Update Service ») avec permissions Application.ReadWrite.All et Mail.Read sur tous les utilisateurs. Cette opération est journalisée dans les logs d'audit Entra ID. **Deuxième alerte Graylog** : règle « Entra ID — Application registration with high-privilege permissions » — criticité HIGH. L'analyste N1 escalade vers le N2.

T+1h15 (10h02) — L'analyste N2 commence l'investigation. Requête Graylog pour corréler l'IP source des connexions Entra avec le baseline habituel de l'administrateur. Résultat : connexion depuis 185.220.101.x (nœud Tor documenté dans l'Abuse.ch database). **Confirmation de compromission.** L'analyste N2 contacte le RSSI.

Cas particuliers

T+1h30 (10h17) — L'attaquant, ayant obtenu des credentials supplémentaires via l'application malveillante, tente une connexion sur l'Active Directory on-premise via un Jump Host exposé sur RDP (port 3389 accessible depuis internet — configuration héritée non documentée). Wazuh détecte la connexion RDP avec l'identifiant de l'administrateur cloud depuis une IP Tor. **Troisième alerte : criticité CRITICAL**. Active Response Wazuh déclenche automatiquement le blocage de l'IP via firewall.

T+2h (10h47) — L'attaquant pivote via un second vecteur : compromission d'un compte de service synchronisé AD Connect (compte « MSOL_xxxxx ») dont le mot de passe n'avait pas été changé depuis 3 ans. Ce compte dispose de permissions DCSync nativement requises par AD Connect. **Alerte Wazuh CRITICAL** : règle DCSync détectée (EventID 4662 sur les attributs réplication AD). Graylog corrèle avec l'anomalie Entra ID détectée précédemment.

T+2h30 (11h17) — Suricata détecte du trafic réseau anormal vers 94.102.49.x (C2 documenté dans les feeds ThreatFox) depuis le Jump Host compromis. Protocole : HTTPS sur port 443, mais certificat auto-signé et JA3 fingerprint correspondant à Cobalt Strike Beacon. **Alerte Suricata CRITICAL**. À ce stade, trois sources indépendantes (Wazuh, Graylog, Suricata) convergent vers le même incident.

T+3h (11h47) — Décision de containment : isolation réseau du Jump Host (VLAN quarantaine), révocation du token de session Entra ID de l'administrateur compromis, désactivation temporaire du compte AD Connect, révocation de l'application Entra malveillante. Le RSSI active la procédure de réponse à incident.

T+4h (12h47) — Forensique rapide sur le Jump Host : image mémoire (Winpmem), analyse avec Volatility3. Cobalt Strike Beacon en mémoire confirmé. Timeline artefacts via Wazuh FIM : aucun fichier créé sur disque — attaque fileless. L'analyste identifie la technique d'injection de processus (T1055.002 — Portable Executable Injection dans svchost.exe).

T+6h (14h47) — Audit complet des logs Graph API des 7 derniers jours sur le compte compromis (rétention Graylog 90 jours). Découverte : l'attaquant avait un accès silencieux depuis 5 jours via une première application malveillante moins visible. Exfiltration de 2340 emails lus via Mail.Read — notification RGPD requise dans les 72h.

T+8h (16h47) — Déclaration NIS2 à l'ANSSI (rapport initial 24h). Notification RGPD à la CNIL préparée. Forensique approfondie confiée à un cabinet PRIS qualifié ANSSI. Rétablissement du service AD Connect via un nouveau compte de service avec mot de passe complexe.

Lessons learned

- Le Jump Host RDP exposé directement sur internet était l'angle mort critique — un inventaire d'actifs rigoureux l'aurait identifié.
- Le compte de service AD Connect avec mot de passe de 3 ans n'était pas suivi dans le gestionnaire de secrets de l'entreprise.
- L'alerte MEDIUM sur l'énumération Graph (T+15min) aurait dû être escaladée immédiatement avec une règle de corrélation « alerte Medium graph + connexion Tor ».
- La détection finale en T+2h30 via Suricata C2 a été le déclencheur du containment effectif — sans NDR, l'incident aurait pu durer des semaines.

Point clé — Cas 1 : La corrélation multi-sources (Wazuh + Graylog + Suricata) a réduit le MTTD (Mean Time To Detect) à 2h30 pour une attaque sophistiquée AiTM + DCSync. Sans corrélation, chaque alerte isolée aurait pu être ignorée. C'est précisément la valeur ajoutée d'une architecture SIEM intégrée vs des outils silotés.

10.2 Cas 2 — Ransomware via RDP Brute-Force, Mimikatz et chiffrement

Contexte et vecteur PME industrielle de 280 employés, secteur manufacturier. Un serveur de fichiers Windows Server 2019 expose RDP sur internet (port 3389) sans VPN — héritage de la période COVID. Le groupe **Akira Ransomware** (actif depuis 2023, double extorsion, affilié à l'ancien réseau Conti) cible ce serveur via une campagne de brute-force distribuée.

Chronologie **J-3 (Vendredi soir) :** Début du brute-force RDP depuis 47 adresses IP distinctes (attaque distribuée évitant le blocage par IP). Volume : 12 000 tentatives en 4 heures. Wazuh génère des alertes de niveau 10 (seuil : 100 tentatives/heure par compte). L'analyste de permanence du week-end (N1 externalisé) note les alertes mais ne les escalade pas — le serveur n'est pas classifié critique dans la CMDB.

J-3 (Samedi 02h17) : Succès du brute-force sur le compte « backup_admin » avec le mot de passe « Backup2023! ». Session RDP établie. Aucune escalade — l'alerte de connexion réussie nocturne génère une alerte MEDIUM classée en observation.

J-2 (Samedi 03h00 à 08h00) : L'attaquant procède à une reconnaissance discrète : commandes net user, net group, nltest /domain_trusts. Exfiltration du fichier ntds.dit via Volume Shadow Copy (T1003.003). Téléchargement de Mimikatz (technique fileless via Invoke-Mimikatz PowerShell encodé en Base64). **Alerte Wazuh CRITICAL** : détection YARA signature Mimikatz en mémoire. Cette alerte, générée à 04h23 du matin un samedi, reste non traitée jusqu'à la prise de service lundi matin.

J-1 (Dimanche) : Mouvement latéral silencieux via Pass-the-Hash avec credentials Mimikatz. Compromission de 3 serveurs supplémentaires. Déploiement du ransomware Akira via GPO modifiée (T1484.001). Wazuh FIM détecte des modifications massives d'extensions de fichiers (.akira ajouté). Volume : 847 alertes FIM en 12 minutes. Active Response Wazuh tente l'isolation réseau mais les GPO ont déjà été propagées.

J0 (Lundi 07h45) : Les employés découvrent les ransom notes. MTTD effectif : 3 jours et demi (alertes générées mais non traitées le week-end).

Réponse et restauration Isolation immédiate de tous les segments réseau affectés. Wazuh Active Response sur les agents encore actifs : coupure réseau, kill des processus de chiffrement en cours. Restauration depuis les sauvegardes Veeam (dernière sauvegarde : 72h avant chiffrement — 3 jours de données perdus). Forensique : chronologie complète reconstituée via les logs Wazuh et Graylog malgré la compromission des serveurs.

Point clé — Cas 2 : La détection technique était excellente (Mimikatz détecté en moins d'une heure). L'échec est organisationnel : pas de procédure d'astreinte weekend pour les alertes CRITICAL. Un SIEM sans processus de réponse 24/7 ne suffit pas — la technologie sans organisation est inefficace. Voir notre fiche réflexe ransomware pour les procédures d'urgence.

10.3 Cas 3 — Supply Chain OAuth : Illicit Consent Grant

Contexte Cabinet de conseil de 120 collaborateurs, Microsoft 365 E3. Un développeur teste une application tierce proposée via un partenaire commercial — outil présenté comme un « assistant

de productivité Microsoft Teams ». L'application demande les permissions OAuth : Mail.ReadWrite, Files.ReadWrite.All, Contacts.Read. Le développeur accorde le consentement sans consulter l'équipe IT.

Exfiltration silencieuse sur 7 jours L'application légitime en apparence masque un comportement malveillant : elle synchronise discrètement les emails et fichiers SharePoint vers un serveur externe. Le volume de données transférées est faible (< 500 MB/jour) — sous les seuils d'alerte DLP configurés. Aucune connexion inhabituelle : l'application utilise les tokens OAuth légitimes accordés par le développeur.

La découverte est **fortuite** : un analyste effectuant une revue mensuelle des applications Entra ID remarque une application inconnue avec des permissions élevées, accordée sans approbation administrative. Il vérifie les logs Graph API pour cette application : 342 appels Mail.Read et 156 appels Files.Read en 7 jours.

Réponse et audit Graph API Révocation immédiate du consentement OAuth via le portail Entra ID. Audit complet via PowerShell (Get-MgOAuth2PermissionGrant) de toutes les applications tierces — découverte de 12 applications supplémentaires avec permissions excessives, dont 3 orphelines (développeur ayant quitté l'entreprise). Analyse des logs Graph API sur 90 jours via Graylog pour évaluer le volume d'exfiltration. Qualification RGPD : emails contenant des données clients → notification CNIL potentiellement requise. Analyse juridique en cours.

Point clé — Cas 3 : L'illicit consent grant est l'une des techniques d'exfiltration cloud les plus difficiles à détecter car elle utilise des mécanismes OAuth légitimes. La prévention passe par une politique stricte d'approbation administrative des applications tierces (Conditional Access + Application Admin Consent Required). La détection post-facto nécessite l'audit régulier des consentements OAuth dans Entra ID, automatisé via Graylog.

10.4 Cas 4 — Insider Threat : Admin sortant, vol de données clients

Contexte Entreprise SaaS B2B de 85 employés. Un administrateur système senior notifie sa démission avec un préavis de 4 semaines. Conformément à la procédure, ses accès sont progressivement réduits. Cependant, aucune surveillance renforcée n'est mise en place durant la période de préavis — lacune organisationnelle fréquente.

Comportements détectés Graylog détecte, lors d'une revue hebdomadaire des activités administratives, plusieurs anomalies comportementales :

- Accès à des dossiers SharePoint hors périmètre habituel (base clients, contrats) — 3x la moyenne historique
- Export PST de sa boîte mail via Outlook (EventID 4663 sur le serveur Exchange — export d'archive)
- Upload de 2,3 GB vers un service cloud personnel (Dropbox) depuis son poste professionnel — détecté via Suricata DPI sur les user-agents Dropbox
- Connexion VPN depuis son domicile à 23h47 un dimanche — hors pattern habituel

La corrélation Graylog de ces 4 signaux faibles, chacun individuellement défendable, constitue un faisceau d'indices solide. L'analyste N2 génère un rapport de comportement anormal (UEBA manuel) sur les 3 semaines précédentes.

Procédure disciplinaire et judiciaire Le RH et le RSSI sont alertés. Les accès de l'administrateur sont révoqués immédiatement (48h avant la fin de préavis officielle). Les logs Graylog constituent des preuves numériques potentielles — ils sont exportés en format immuable (PDF signé électroniquement + hashage SHA-256) et conservés dans un coffre-fort numérique. Un prestataire forensique est mandaté pour l'analyse du poste professionnel. La procédure prud'homale et, potentiellement, pénale (violation du secret des affaires, Art. L151-1 Code de commerce) est engagée.

Point clé — Insider Threat : Les menaces internes nécessitent une approche **UEBA** (User and Entity Behavior Analytics) basée sur la corrélation comportementale sur le temps long. Wazuh et Graylog seuls ne suffisent pas — il faut configurer des dashboards de baseline comportementale par utilisateur et des alertes sur les écarts significatifs. Des outils dédiés (Exabeam, Microsoft Insider Risk Management) complètent utilement la stack pour ce cas d'usage.

11. Perspectives et évolutions

Le paysage des outils SIEM open source évolue à un rythme soutenu, porté à la fois par la maturité croissante des projets existants et par l'émergence de nouvelles approches architecturales. Cette section dresse un panorama des évolutions attendues à horizon 2026-2027 pour chaque composant de la stack, ainsi que pour les technologies complémentaires qui redéfinissent le périmètre du SOC moderne.

11.1 Wazuh 5.x — Roadmap et nouveautés attendues

Wazuh 4.x a marqué une rupture significative avec les versions précédentes en remplaçant la dépendance à Elasticsearch par OpenSearch et en introduisant le tableau de bord intégré. La roadmap Wazuh 5.x, communiquée partiellement lors du Wazuh Summit 2024, s'articule autour de trois axes majeurs.

Le premier axe concerne l'**architecture distribuée native** : Wazuh 5.x vise à rompre le modèle monolithique du manager unique, goulot d'étranglement pour les très grands déploiements. Une architecture à base de composants indépendants (API server, event processor, indexer connector) permettra un scaling horizontal véritable, ouvrant la voie à des déploiements de plusieurs millions d'agents sans architecture cluster complexe.

Le second axe porte sur l'**enrichissement contextuel automatique** : intégration native de sources OSINT (Shodan, VirusTotal, AbuseIPDB) sans configuration manuelle, corrélation automatique avec les CVE NIST NVD pour les alertes de vulnérabilité, et contextualisation des assets via des connecteurs CMDB standardisés.

Le troisième axe concerne le **module UEBA natif** : Wazuh 5.x devrait intégrer des capacités d'analyse comportementale de base (baseline automatique par utilisateur, détection d'anomalies statistiques) réduisant la dépendance à des outils tiers pour les cas d'usage insider threat les plus courants.

11.2 Graylog 6.x et l'évolution OpenSearch

Graylog 6.x (disponible en bêta fin 2024, GA attendu mi-2025) introduit des changements architecturaux importants. La dépendance à MongoDB est progressivement réduite au profit d'une architecture orientée événements avec Apache Kafka comme bus de messages optionnel — améliorant la résilience en cas de pic d'ingestion.

L'interface utilisateur est intégralement réécrite en React avec une approche mobile-first et des capacités de personnalisation de dashboard avancées. Les **Security Views** (vues de sécurité préconfigurées) intègrent nativement des requêtes orientées MITRE ATT&CK, réduisant le temps de configuration initiale pour les équipes SOC.

Du côté d'**OpenSearch** (fork Apache 2.0 d'Elasticsearch par AWS), la version 3.x introduit le support natif des vecteurs haute dimension (k-NN search), ouvrant la voie à des recherches sémantiques dans les logs — une capacité que les moteurs de recherche lexicaux traditionnels ne peuvent offrir. Cette évolution est structurante pour l'intégration des LLMs dans le workflow SOC.

Point clé — Graylog 6.x : La migration de Graylog 5.x vers 6.x nécessitera une planification soigneuse — les formats de données MongoDB ont changé et la migration des dashboards existants n'est pas automatique. Anticipez 2-3 jours de travail de migration pour un environnement de taille moyenne. Testez en environnement de recette avant toute mise à jour en production.

11.3 Suricata — eBPF, DPDK et détection ML native

Suricata 7.x (sorti en 2024) et la roadmap 8.x (prévue 2025-2026) introduisent des évolutions majeures sur les performances et les capacités de détection. eBPF (Extended Berkeley Packet Filter) permet à Suricata de capturer les paquets directement dans le kernel Linux sans copie en espace utilisateur — réduisant la latence de capture de 40 à 70% selon les benchmarks OISF (Open Information Security Foundation).

DPDK (Data Plane Development Kit) pousse cette logique encore plus loin : en byppassant complètement la pile réseau Linux pour la capture, des débits de 40-100 Gbps deviennent atteignables sur du matériel standard. Cette capacité est critique pour les environnements datacenter haute densité où les sondes réseau traditionnelles créent des goulots d'étranglement.

La détection **ML native** dans Suricata est l'évolution la plus structurante : l'intégration du projet *Suricata Machine Learning* (SML) permet d'entraîner des modèles de classification de trafic directement dans Suricata, sans externalisation vers un outil tiers. Les premiers cas d'usage cibles sont la détection de tunnels DNS (exfiltration via requêtes DNS), la classification des flux chiffrés (TLS fingerprinting avancé au-delà de JA3/JA3S) et la détection de scan réseau lent (slow scan évitant les seuils de volume).

11.4 IA et LLMs dans le SOC — Révolution ou gadget ?

L'irruption des **Large Language Models** dans les outils de cybersécurité est le sujet dominant des conférences SOC en 2024-2025 (RSA, Black Hat, FIC). La question n'est plus « est-ce que l'IA va changer le SOC ? » mais « quels cas d'usage sont réellement opérationnels aujourd'hui et lesquels relèvent encore du marketing ? ».

Triage et résumé d'alertes : C'est le cas d'usage le plus mature. Des outils comme Microsoft Copilot for Security, Google Security AI et les versions entreprise de Claude/GPT-4 peuvent ingérer une alerte SIEM et produire en quelques secondes un résumé structuré : contexte de l'asset concerné, technique MITRE mappée, risque estimé, actions recommandées. Wazuh 4.9+ expérimente une intégration LLM via son module d'analyse contextuelle. Le gain de temps pour les analystes N1 est mesurable : 30 à 50% de réduction du temps de triage selon les pilotes documentés par des équipes SOC européennes en 2024.

Threat hunting assisté par le langage naturel : L'interface *Natural Language → Query* (NL2Q) permet à un analyste de poser une question en français ou en anglais et d'obtenir la requête Graylog/OpenSearch/Sigma correspondante. Exemple : « Montre-moi toutes les connexions RDP réussies depuis des IP étrangères vers des serveurs DC entre minuit et 6h du matin cette semaine » → requête OpenSearch générée automatiquement. Cette capacité democratise l'accès au threat hunting pour les analystes N1 moins expérimentés.

Chatbot SOC — Procédures et IOCs : Un assistant conversationnel alimenté par les runbooks de l'organisation, les bases d'IOCs et les historiques d'incidents permet à un analyste N1 en astreinte de nuit d'obtenir instantanément la procédure à suivre face à un type d'alerte spécifique. Implémenté via RAG (Retrieval-Augmented Generation) sur la base documentaire interne, ce chatbot ne remplace pas le jugement humain mais réduit significativement le stress des analystes juniors en situation d'urgence.

Point clé — LLMs dans le SOC : Les hallucinations des LLMs sont un risque réel dans le contexte SOC — un faux négatif (« cette alerte n'est pas critique ») ou un faux positif (« bloquer immédiatement cet asset ») générés par un modèle peut avoir des conséquences opérationnelles graves. Toujours maintenir un humain dans la boucle de décision. Utilisez les LLMs pour *assister* l'analyste, jamais pour *remplacer* son jugement.

11.5 Vers une stack XDR complète

Le SIEM hybride décrit dans ce guide est le socle. L'évolution naturelle vers un **XDR** (Extended Detection and Response) complet intègre des couches supplémentaires de visibilité et de réponse :

- **Velociraptor** : plateforme open source de réponse à incident et de forensique live. Complémentaire à Wazuh pour les investigations approfondies — collecte d'artefacts à la demande, exécution de requêtes VQL sur l'ensemble du parc, timeline d'activité forensique.
- **Osquery** : expose le système d'exploitation comme une base de données SQL. Permet des requêtes ponctuelles ou continues sur l'état des endpoints (processus, connexions réseau, fichiers, registry). Intégration Graylog via osquery logger plugin.
- **Zeek (anciennement Bro)** : NDR (Network Detection and Response) de référence pour l'analyse de trafic réseau en profondeur. Complémentaire à Suricata : là où Suricata applique des signatures, Zeek génère des logs structurés de haute fidélité (connexions, HTTP, DNS, TLS, SMB) permettant des analyses comportementales sans signature.
- **CASB** (Cloud Access Security Broker) : visibilité et contrôle sur l'utilisation des applications cloud. Pour les environnements M365, Microsoft Defender for Cloud Apps (anciennement MCAS) intègre nativement les logs dans Sentinel — ou peut exporter vers Graylog via API.
- **SOAR — Shuffle / Tines** : les plateformes SOAR (Security Orchestration Automation and Response) automatisent les workflows de réponse. Shuffle (open source) ou Tines (freemium) s'intègrent avec

Graylog et Wazuh via webhooks et API REST pour orchestrer des playbooks de réponse complexes sans intervention humaine pour les cas les plus courants.

11.6 Zero Trust Architecture

Le **NIST SP 800-207** définit la Zero Trust Architecture (ZTA) comme un paradigme de sécurité qui déplace la défense périmétrique vers une évaluation continue de chaque accès, indépendamment de la localisation réseau de l'utilisateur ou de l'appareil. La stack SIEM s'inscrit naturellement dans une ZTA comme composant de *vérification continue*.

Dans une architecture Zero Trust, le SIEM reçoit les signaux des composants Policy Enforcement Points (PEP) et Policy Decision Points (PDP) — Entra ID Conditional Access, Zscaler, Cloudflare Access — et les corrèle avec les événements endpoint et réseau pour calculer un **score de confiance dynamique**. Un score dégradé (anomalie comportementale détectée) peut déclencher automatiquement une réévaluation de l'accès via le Policy Engine, sans intervention humaine.

Point clé — Zero Trust + SIEM : Le SIEM n'est pas optionnel dans une architecture Zero Trust — il est le cerveau analytique qui informe les décisions d'accès dynamiques. La bidirectionnalité est clé : le SIEM reçoit les logs ZTA, mais envoie aussi des signaux de risque aux composants ZTA pour adapter les politiques d'accès en temps réel.

12. Conclusion

12.1 Synthèse des bénéfices

Au terme de ce guide en trois parties, la stack **Wazuh + Graylog + Suricata + intégration AD/M365** démontre sa capacité à couvrir l'ensemble de la *kill chain* de Lockheed Martin et du framework MITRE ATT&CK avec des outils 100% open source, souverains et maîtrisés.

La **couverture kill chain** est multidimensionnelle : Suricata intercepte les phases de reconnaissance réseau et de command & control (NDR) ; Wazuh couvre les techniques d'exécution, de persistence, d'élévation de privilèges et d'impact (EDR) ; Graylog corrèle l'ensemble et offre la visibilité sur les mouvements latéraux et l'accès aux credentials (SIEM). Cette complémentarité fonctionnelle sans recouvrement inutile est le principal avantage architectural de la stack.

La **souveraineté numérique** est un bénéfice stratégique sous-estimé. Vos logs de sécurité ne transitent pas par des serveurs américains ou asiatiques. Vous connaissez exactement le code qui analyse vos événements. En cas de CVE critique sur un composant, vous pouvez appliquer le correctif en quelques heures sans attendre un éditeur. Cette maîtrise de la chaîne de dépendances est particulièrement précieuse pour les OIV, les entités NIS2 et les organisations traitant des informations classifiées.

Le **TCO maîtrisé** est réel à partir de 500 endpoints : économies de licence de 200k€ à 2M€ sur 3 ans selon la taille de l'organisation, au prix d'un investissement en compétences internes qui renforce durablement l'équipe SOC. La **communauté** mondiale Wazuh (60 000+ membres en 2025), Graylog et Suricata offre un niveau de support pair-à-pair souvent supérieur au support contractuel des éditeurs propriétaires pour les problèmes techniques courants.

12.2 Quand choisir cette stack versus une solution propriétaire

Table 15: Guide de décision — Stack open source vs propriétaire

Critère	Open Source (cette stack)	Propriétaire (Splunk/Sentinel/QRadar)
Budget IT annuel sécurité	< 500k€ — avantage open source	> 500k€ — propriétaire viable
Équipe SOC interne	≥ 1 ETP dédié requis	Peut fonctionner avec 0,5 ETP + support édit
Souveraineté exigée	Recommandé fortement	Risqué (cloud US, FISA 702)
Personnalisation avancée	Illimitée (accès code source)	Limitée aux APIs et plugins certifiés
Conformité rapide (audit immédiat)	Délai 3-6 mois de déploiement	Délai 1-2 mois (SaaS)
Volume > 100 GB/jour	Nécessite expertise OpenSearch	Splunk scalable nativement (coûteux)
Support 24/7 garanti (SLA)	Wazuh Enterprise ou SOC managé	SLA contractuel avec l'éditeur

La stack open source décrite dans ce guide est le choix optimal pour les organisations qui combinent trois caractéristiques : budget limité (sous 300k€/an de budget sécurité total), exigences de souveraineté ou de personnalisation, et capacité à maintenir des compétences internes. Pour les grandes entreprises disposant de budgets importants et d'une exigence de support contractuel garanti, les solutions propriétaires restent pertinentes — en particulier Microsoft Sentinel pour les environnements fortement M365, qui bénéficie d'une intégration native incomparable avec la suite Microsoft.

12.3 Ressources et formations

La montée en compétences sur cette stack nécessite une combinaison de formation formelle et de pratique. Les ressources clés :

- **Formation Wazuh officielle** : Wazuh Training (wazuh.com/training) — parcours de 40 heures couvrant déploiement, règles, réponse à incident.
- **Graylog University** : cours en ligne gratuits sur la configuration, les pipelines et les dashboards.
- **OISF Suricata Training** : formations officielles sur les règles, les performances et l'analyse de logs.
- **SANS FOR572** : Network Forensics — référence mondiale sur l'analyse de trafic réseau et les SIEM.
- **Blue Team Labs Online / TryHackMe** : laboratoires pratiques sur les scénarios de détection SIEM.
- **MITRE ATT&CK Navigator** : outil indispensable pour visualiser la couverture de détection de votre stack.
- **Guide ANSSI — Recommandations de sécurité pour la journalisation** : référence réglementaire française incontournable.

12.4 Services Ayi NEDJIMI Consultants

La complexité de déploiement et d'exploitation d'une stack SIEM hybride de niveau expert justifie souvent un accompagnement par des spécialistes ayant déjà parcouru ce chemin. **Ayi NEDJIMI Consultants**

propose un accompagnement complet sur l'ensemble du cycle de vie de votre SIEM open source :

- **Architecture et conception** : dimensionnement adapté à votre volumétrie, choix des composants, définition des flux de données, intégration dans votre architecture réseau et cloud existante. Livrable : schéma d'architecture détaillé + dossier de conception technique.
- **Déploiement et configuration** : installation sécurisée de l'ensemble de la stack, hardening des composants selon les CIS Benchmarks, configuration des sources de logs (AD, M365, équipements réseau, applicatifs), création des règles de détection initiales mappées sur votre contexte métier.
- **Run SOC / SIEM managé** : exploitation quotidienne de votre SIEM par nos analystes, triage des alertes, escalade des incidents confirmés, rapports mensuels de posture sécurité, mise à jour continue des règles de détection face aux nouvelles menaces.
- **Audit de maturité SOC** : évaluation de votre SIEM existant par rapport aux meilleures pratiques (CIS Controls, NIST CSF, ISO 27001), identification des angles morts de détection via Red Team ciblé, recommandations priorisées avec plan de remédiation.
- **Formation interne** : programmes sur mesure pour vos équipes (analystes N1/N2, ingénieurs SIEM, RSSI) — de la prise en main de l'interface Graylog à la rédaction de règles Sigma avancées.

Retrouvez nos guides d'expertise sur la checklist ISO 27001 Annexe A et notre portail de réponse à incident pour approfondir votre préparation opérationnelle.

Conclusion : La stack Wazuh + Graylog + Suricata n'est pas la solution la plus simple à déployer, ni la plus rapide à opérationnaliser. Mais c'est aujourd'hui la combinaison la plus puissante, la plus souveraine et la plus économique pour les organisations qui ont les ressources humaines pour l'exploiter correctement. Dans un paysage de menaces où LockBit 4.0, BlackCat ALPHV reborn, Cl0p, Akira et Play continuent d'innover, la question n'est pas « avons-nous besoin d'un SIEM ? » mais « quel niveau de visibilité et de réactivité voulons-nous atteindre ? ». Ce guide vous a donné les bases pour répondre ambitieusement à cette question.

Annexes

Annexe A — Référentiel de règles personnalisées

Les règles présentées dans cette annexe constituent un point de départ opérationnel, non un catalogue exhaustif. Chaque règle doit être testée et validée dans votre environnement avant déploiement en production. Les seuils (fréquences, compteurs) sont indicatifs et doivent être ajustés selon votre baseline observée.

A.1 — Règles Wazuh XML (5 règles) Les règles Wazuh sont définies en XML dans les fichiers `/var/ossec/etc/rules/local_rules.xml`. Le niveau va de 0 (informatif) à 15 (critique maximal). Les règles ci-dessous couvrent les techniques d'attaque les plus fréquemment observées en 2024-2025.

Notes complémentaires

```
<!-- Règle A.1.1 : Kerberoasting Detection -->
<!-- Technique MITRE ATT&CK : T1558.003 - Steal or Forge Kerberos Tickets: Kerberoasting -->
<!-- Détecte les demandes de tickets TGS pour des comptes de service (SPN) -->
<!-- Source : Windows Security EventID 4769, Ticket Encryption Type 0x17 (RC4) -->
<group name="windows,kerberos,credential_access">
# ... (extrait — voir documentation officielle)
```

A.2 — Règles Sigma YAML avec conversion Wazuh Sigma est un format de règle de détection générique, agnostique de la plateforme SIEM. L'outil `sigma-cli` avec le backend Wazuh convertit les règles Sigma en règles Wazuh XML nativement depuis 2024. Cette approche permet de partager des règles dans l'écosystème communautaire (SigmaHQ GitHub, 3000+ règles) et de les adapter à votre stack.

```
# Règle Sigma A.2.1 : Détection Pass-the-Hash via WMI
# Source : https://github.com/SigmaHQ/sigma
# Référence ATT&CK : T1550.002
title: Pass-the-Hash via WMI Remote Execution
id: a6b49e40-1c02-4c8e-9d4b-7d3a2f8e5c6f
# ... (extrait — voir documentation officielle)
```

```
# Règle Suricata A.3.1 : Détection C2 générique (beacon régulier)
# Détecte des connexions sortantes à intervalles réguliers (±5% variation)
# caractéristiques d'un beacon C2 (Cobalt Strike, Sliver, Brute Ratel)
alert tcp $HOME_NET any -> $EXTERNAL_NET any (
    msg:"ET CUSTOM C2 Beacon Pattern - Reguler Interval HTTPS";
# ... (extrait — voir documentation officielle)
```

A.3 — Règles Suricata personnalisées **Point clé — Règles personnalisées :** Toute règle personnalisée doit être versionnée dans un dépôt Git dédié (ex. `siem-rules`) avec un processus de revue de code avant déploiement. Une règle mal rédigée peut générer des milliers de faux positifs par heure et saturer votre équipe SOC. Testez systématiquement avec des captures PCAP de référence avant mise en production.

Annexe B — Structures de déploiement (références architecturales)

Cette annexe présente les structures de déploiement sans détailler les valeurs de configuration spécifiques à votre environnement. Ces structures servent de guide architectural pour vos équipes d'infrastructure.

```
# Structure référentielle docker-compose.yml — Stack SIEM
# NE PAS utiliser tel quel en production — adapter à votre environnement
version: '3.8'

services:
# ... (extrait — voir documentation officielle)
```

B.1 — Structure docker-compose.yml

```
# Structure site.yml — Playbook Ansible SIEM
# Organisation des rôles par fonction

# site.yml (orchestrateur principal)
---
# ... (extrait — voir documentation officielle)
```

B.2 — Structure Playbook Ansible

```
# Structure Kubernetes pour déploiement cloud-native (optionnel)
# Namespace dédié : siem-system

# Hiérarchie des manifests :
# k8s/
# ... (extrait — voir documentation officielle)
```

B.3 — Structure Manifests Kubernetes **Point clé — Kubernetes SIEM :** Le déploiement d'un SIEM sur Kubernetes introduit une complexité opérationnelle significative (gestion des volumes persistants, NetworkPolicies, secrets management via Vault/Sealed Secrets). Recommandé uniquement si votre organisation dispose d'une équipe Platform Engineering mature. Pour la majorité des cas, un déploiement sur VMs avec Ansible est plus robuste et plus facile à maintenir.

Annexe C — Runbooks SOC

Les runbooks suivants sont des procédures opérationnelles destinées aux analystes SOC. Ils doivent être accessibles hors ligne (PDF imprimé ou application mobile dédiée) car lors d'un incident majeur, l'accès aux systèmes internes peut être compromis. Chaque runbook inclut les décisions clés, les outils utilisés et les escalades prévues.

C.1 — Runbook : Investigation logon anormal **Déclencheur :** Alerte Wazuh/Graylog de niveau ≥ 10 sur une connexion (EventID 4624, 4625, 4648) avec au moins un des critères : heure inhabituelle, localisation géographique anormale, compte privilégié, nombre de tentatives élevé.

Étape 1 — Qualification initiale (5 minutes maximum) :

- Identifier l'utilisateur concerné : est-il en poste actuellement ? Y a-t-il une mission déplacement/télétravail connue ?
- Identifier l'IP source : géolocalisation (MaxMind GeoIP dans Graylog), réputation (AbuseIPDB, VirusTotal), type (VPN connu, Tor, datacenter, résidentiel).
- Identifier la cible : quel serveur/service ? Quel niveau de criticité selon la CMDB ?
- Décision rapide : **faux positif probable** (utilisateur en déplacement connu, IP d'un VPN corporate) → documenter et clore. **Suspect** → Étape 2.

Étape 2 — Investigation Graylog (15 minutes) :

- Requête Graylog : tous les événements du compte sur les 7 derniers jours, tri chronologique. Identifier les patterns habituels (heures de connexion, IP sources, services accédés).
- Rechercher les EventID 4720 (création compte), 4732 (ajout groupe), 4728 (ajout groupe global) dans les logs AD récents pour ce compte.
- Vérifier les logs M365 (Unified Audit Log via Graylog) : applications OAuth autorisées récemment, règles inbox créées, partages SharePoint récents.
- Corréler avec Suricata : trafic réseau inhabituel depuis/vers l'IP de la session suspecte dans la même fenêtre temporelle.

Étape 3 — Décision et escalade :

- **Compromission probable** : escalader vers N2 immédiatement, ne pas alerter l'utilisateur (risque de tip-off si insider). Activer le runbook C.3 si MFA fatigue, ou le runbook C.2 si activité ransomware détectée.
- **Compromission confirmée** : décision de containment (N2 ou RSSI selon criticité). Documenter toutes les actions dans le système de ticketing avec horodatage.
- **Faux positif confirmé** : documenter la justification, envisager l'ajout d'une exception CMDB si récurrent.

Étape 4 — Containment si déclenché :

Stratégie de déploiement

- Révoquer les sessions Entra ID actives (PowerShell : Revoke-MgUserSignInSession).
- Désactiver le compte AD (Set-ADUser -Enabled $false) — avec validation RSSI pour les comptes critiques.
- Bloquer l'IP source via ACL firewall ou Conditional Access Entra.
- Changer le mot de passe du compte (ne pas réactiver avant investigation complète).
- Notifier le responsable hiérarchique de l'utilisateur.

C.2 — Runbook : Réponse ransomware **Déclencheur :** Alerte Wazuh niveau 15 FIM (extensions connues ransomware) OU découverte manuelle de fichiers chiffrés OU ransom note trouvée.

Phase 1 — Containment (objectif : 15 minutes)

- **T+0** : Confirmer l'alerte. Ne PAS redémarrer les machines affectées (risque de perte de preuves mémoire et d'effacement des shadow copies).
- **T+2min** : Activer l'Active Response Wazuh pour isoler réseau des endpoints affectés. En parallèle, isolation manuelle si nécessaire (déconnexion câble réseau, désactivation Wi-Fi).
- **T+5min** : Identifier le patient zéro via Graylog (premier endpoint avec alertes FIM ransomware). Remonter la timeline 24-48h avant sur ce poste.
- **T+10min** : Isoler les segments réseau adjacents aux endpoints confirmés. Suspendre les réplications AD et les sauvegardes automatiques (pour éviter d'écraser les backups sains avec des backups chiffrés).
- **T+15min** : Notification RSSI, DG, et si EE/EI NIS2 : préparation rapport 24h ANSSI.

Phase 2 — Eradication (objectif : 4-8 heures)

- Image forensique des endpoints compromis (Winpmem pour RAM, FTK Imager pour disque) avant tout nettoyage.
- Identification du vecteur d'intrusion initial via Graylog/Wazuh (RDP brute-force ? Phishing ? Vulnérabilité appliquée ?). **Sans connaître le vecteur, la reinfection est certaine.**
- Identifier et fermer le vecteur (patch, changement règle firewall, reset credentials compromis).
- Scanner l'ensemble du parc avec Wazuh YARA (règles ransomware) pour identifier d'autres machines potentiellement compromises mais non chiffrées.
- Réinitialisation des credentials compromis (krbtgt si DCSync, tous les comptes de service si Mimikatz confirmé).

Phase 3 — Recovery (objectif : 24-72 heures)

- Restauration depuis les sauvegardes les plus récentes antérieures à l'intrusion (identifier la date grâce à la timeline forensique).
- Validation des sauvegardes avant restauration (intégrité, absence de malware dans les backups eux-mêmes).
- Redémarrage progressif des services par ordre de priorité métier, avec surveillance renforcée Wazuh/Graylog.
- Communication aux utilisateurs : message officiel de la direction, consignes de vigilance, hotline.

C.3 — Runbook : Investigation MFA Fatigue **Déclencheur :** Règle Wazuh 100231 (5+ refus MFA en 5min pour le même compte).

- **Étape 1** : Contacter immédiatement l'utilisateur par téléphone (pas par email — le compte est peut-être compromis). Lui demander : « Avez-vous reçu des notifications MFA que vous n'avez pas demandées ? » Si oui → compromission probable.
- **Étape 2** : Même si l'utilisateur dit n'avoir rien accepté, vérifier les logs Entra : y a-t-il eu une connexion réussie après les refus ? Si oui → l'utilisateur a peut-être accepté par erreur ou l'attaquant a utilisé une technique alternative (session cookie volé, MFA bypass).
- **Étape 3** : Révoquer toutes les sessions actives Entra ID du compte (Revoke-MgUserSignInSession). Forcer un changement de mot de passe.
- **Étape 4** : Vérifier l'IP source des tentatives MFA : si IP Tor ou VPN anonyme → campanhe d'attaque ciblée. Rechercher d'autres comptes ayant reçu des tentatives similaires depuis la même IP dans les 48h.
- **Étape 5** : Recommander à l'utilisateur de passer sur une méthode MFA résistante au phishing (FIDO2/clé de sécurité physique) plutôt que les push notifications.

C.4 — Runbook : Vérification post-incident À réaliser systématiquement dans les 48-72 heures suivant la clôture de tout incident de niveau ≥ 3 (criticité élevée).

- **Vérification technique** : Confirmer que le vecteur d'intrusion est fermé. Scanner le périmètre avec l'outil de détection correspondant (Wazuh YARA pour malware, nmap pour ports ouverts, audit OAuth

pour illicit grant). Vérifier que les règles de détection ont bien fonctionné et identifier les éventuels angles morts.

- **Vérification des logs** : S'assurer que la chaîne de journalisation est intacte et que les logs de l'incident sont conservés dans un stockage immuable (pour usage judiciaire éventuel). Vérifier l'intégrité des logs (hash SHA-256, signature électronique).
- **Post-mortem** : Réunion obligatoire avec les parties prenantes dans les 5 jours ouvrés. Format blameless post-mortem (axé sur les processus, pas les individus). Documenter : timeline, actions prises, décisions et leur justification, lessons learned, actions correctives avec propriétaire et date.
- **Mise à jour des règles** : Si une technique d'attaque n'a pas été détectée, créer la règle Wazuh/Suricata correspondante et tester sa rétrocompatibilité avec les logs de l'incident.
- **Mise à jour des runbooks** : Si la procédure a révélé des lacunes, mettre à jour les runbooks et former l'équipe sur les changements.
- **Métriques** : Calculer et documenter le MTTD (Mean Time To Detect) et le MTTR (Mean Time To Respond) de l'incident pour le suivi de performance du SOC.

Point clé — Runbooks : Un runbook non testé est un runbook qui échouera en production. Organisez des exercices de simulation (table top) trimestriels où l'équipe parcourt les runbooks sur des scénarios fictifs. Mettez à jour les runbooks après chaque incident réel et chaque simulation. Les runbooks doivent vivre dans un système de gestion documentaire versionné (Confluence, Notion, ou simplement Git).

Annexe D — Glossaire technique

Les termes suivants sont définis dans leur contexte SIEM/SOC. Certains ont des définitions plus larges dans d'autres domaines informatiques.

Table 16: Glossaire technique — 3

Terme	Développé	Définition
SIEM	Security Information and Event Management	Plateforme centralisant la collecte, la corréla
XDR	Extended Detection and Response	Évolution du EDR/NDR/SIEM qui unifie la dé
EDR	Endpoint Detection and Response	Solution de sécurité installée sur les endpoi
NDR	Network Detection and Response	Solution analysant le trafic réseau pour déte
UEBA	User and Entity Behavior Analytics	Analyse comportementale des utilisateurs e
FIM	File Integrity Monitoring	Surveillance en temps réel des modification
C2	Command and Control	Infrastructure utilisée par un attaquant pour
IOC	Indicator of Compromise	Artefact observable (IP, hash, domaine, URL
TTP	Tactics, Techniques and Procedures	Comportements et méthodes utilisés par les
EPS	Events Per Second	Métrique de volumétrie de logs. Indicateur c
ECS	Elastic Common Schema	Standard de normalisation des champs de lo

Terme	Développé	Définition
CIM	Common Information Model	Schéma de normalisation de données dével
SOAR	Security Orchestration, Automation and Response	Plateforme automatisant les workflows de r
MTTD	Mean Time To Detect	Temps moyen entre le début d'une attaque
MTTR	Mean Time To Respond	Temps moyen entre la détection d'un incide
APT	Advanced Persistent Threat	Acteur malveillant sophistiqué (souvent état
RBAC	Role-Based Access Control	Modèle de contrôle d'accès où les permissio
YARA	Yet Another Recursive Acronym	Langage de règles pour identifier et classifie
Sigma	—	Format de règle de détection générique, agr
MISP	Malware Information Sharing Platform	Plateforme open source de partage de threa
CASB	Cloud Access Security Broker	Composant de sécurité assurant la visibilité
PASSI	Prestataires d'Audit de la Sécurité des Systèmes d'Information	Label de qualification ANSSI pour les prestat
SOC	Security Operations Center	Centre opérationnel dédié à la surveillance,
PHS	Password Hash Synchronization	Mode de synchronisation AD Connect (Entra
AiTM	Adversary-in-the-Middle	Technique de phishing avancée utilisant un
DCSync	—	Technique d'attaque (T1003.006) exploitant
SBOM	Software Bill of Materials	Inventaire exhaustif des composants logicie
WORM	Write Once Read Many	Technologie de stockage immuable où les d
ZTA	Zero Trust Architecture	Paradigme de sécurité (NIST 800-207) élimin
GELF	Graylog Extended Log Format	Format de messages JSON compressé dével
RAG	Retrieval-Augmented Generation	Architecture LLM combinant un moteur de r

Annexe E — Références utiles

Les ressources suivantes constituent la bibliographie de référence pour la mise en œuvre et l'évolution de la stack SIEM décrite dans ce guide.

Table 17: Références documentaires clés

Catégorie	Ressource	URL
Documentation officielle	Wazuh Documentation	documentation.wazuh.com
Documentation officielle	Graylog Documentation	go2docs.graylog.org
Documentation officielle	Suricata User Guide	suricata.readthedocs.io

Catégorie	Ressource	URL
Framework	MITRE ATT&CK Navigator	mitre-attack.github.io/attack-navigator
Framework	NIST CSF 2.0	nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29
Framework	NIST SP 800-207 Zero Trust	nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800
Benchmarks	CIS Benchmarks Ubuntu 24.04	cisecurity.org/benchmark/ubuntu_linux
Réglementation FR	ANSSI — Guide journalisation	ssi.gouv.fr
Réglementation UE	ENISA NIS2 Guidelines	enisa.europa.eu
Règles détection	SigmaHQ GitHub	github.com/SigmaHQ/sigma
Règles détection	YARA-Rules GitHub	github.com/Yara-Rules/rules
Règles réseau	Emerging Threats (ET) Open	rules.emergingthreats.net
Threat Intelligence	MISP Project	misp-project.org
Threat Intelligence	Abuse.ch Threat Feeds	abuse.ch
Formation	Wazuh Training	wazuh.com/training
Labs pratiques	Blue Team Labs Online	blueteamlabs.online
Normes	ISO 27001:2022	iso.org/standard/82875.html

Annexe F — Matrice de compatibilité des versions

Le maintien de la compatibilité inter-composants est l'un des défis opérationnels majeurs d'une stack open source. Cette matrice liste les combinaisons testées et validées en production ou en laboratoire. Les versions marquées **EOL** (End of Life) ne reçoivent plus de correctifs de sécurité et doivent être mises à jour en priorité.

Table 18: Matrice de compatibilité — Versions validées (mai 2025)

Wazuh	OpenSearch (Indexer)	Graylog	OpenSearch (Graylog)	MongoDB	OS Recommandé	Statut
4.9.x	2.13.x	5.2.x	2.11.x	7.0.x	Ubuntu 24.04 LTS	**Recommandé Pr**
4.8.x	2.11.x	5.1.x	2.10.x	6.0.x	Ubuntu 22.04 LTS	Stable, supporté
4.7.x	2.8.x	5.0.x	2.8.x	5.0.x	Ubuntu 22.04 LTS	Maintenance uniq
4.6.x	2.6.x	4.3.x	2.5.x	4.4.x	Ubuntu 20.04 LTS	EOL approchant –
4.5.x	2.4.x	4.2.x	2.3.x	4.4.x	Ubuntu 20.04 LTS	EOL — ne pas utili

Table 19: Matrice de compatibilité Suricata — Versions et fonctionnalités

Suricata	Kernel Linux min.	eBPF support	DPDK support	Débit max testé	Statut
7.0.x	5.15+	Oui (complet)	Oui (expérimental)	25 Gbps	**Recommandé Production**
6.0.x	4.19+	Partiel	Non	10 Gbps	Maintenance LTS
5.0.x	4.15+	Non	Non	5 Gbps	EOL — ne pas utiliser

Point clé — Gestion des versions : Planifiez vos mises à jour SIEM 3 mois à l'avance. Testez systématiquement en environnement de staging avant toute mise à jour majeure en production. Les incompatibilités de version les plus fréquentes concernent MongoDB (changement de format de données entre versions majeures) et les pipelines Graylog (syntaxe modifiée entre 4.x et 5.x). Maintenez un runbook de rollback pour chaque composant.

Limites honnêtes de la stack open source : Soyons transparents sur ce que cette stack ne fait pas aussi bien que les solutions propriétaires haut de gamme. L'**UEBA native** (Exabeam, Microsoft Insider Risk Management) est plus sophistiquée que ce que Graylog et Wazuh peuvent offrir sans configuration extensive. Le **support SOAR intégré** (Splunk SOAR, IBM QRadar SOAR) est plus mature que Shuffle ou Tines open source. La **gestion des incidents multitenants** (pour les MSSPs gérant plusieurs clients) est plus complexe avec cette stack qu'avec des plateformes dédiées comme Devo ou Logpoint. Et la **courbe d'apprentissage** est significativement plus élevée — comptez 6 à 12 mois pour qu'une équipe atteigne un niveau de maîtrise opérationnelle suffisant. Ces limites sont réelles et doivent être intégrées dans votre décision d'adoption.

Questions fréquentes

Combien de temps faut-il pour déployer cette stack SIEM en production ?

Pour un environnement de 500 endpoints, comptez 3 à 4 semaines pour le déploiement initial (architecture, hardening, intégrations AD/M365), puis 6 à 8 semaines de tuning des règles avant un fonctionnement SOC pleinement opérationnel.

Quel est le coût total de possession sur 3 ans ?

Pour 500 endpoints, comptez environ 25 000 € à 40 000 € par an (matériel + administration + formation), soit 5 à 10 fois moins qu'une solution propriétaire équivalente comme Splunk ou Microsoft Sentinel.

Cette stack est-elle conforme NIS2 et ISO 27001:2022 ?

Oui, à condition d'être correctement déployée. Wazuh fournit nativement les mappings de conformité, et la stack permet de répondre aux exigences A.5.7, A.8.15, A.8.16 d'ISO 27001:2022 ainsi qu'aux obligations de notification 24h/72h de NIS2.