Zabbix 7 en 2026 : Supervision Sécurité et Alertes Avancées

> Guide Zabbix 7 pour la supervision de sécurité en 2026 — templates sécurité, alertes comportementales, intégration SIEM, chiffrement PSK/TLS et HA cluster.

Zabbix est l'outil de supervision infrastructure open source le plus déployé en Europe francophone, présent dans des milliers d'entreprises françaises, de collectivités territoriales et d'administrations publiques de toute taille. La version 7.0 LTS, publiée en 2024 et maintenue jusqu'en 2029, apporte des améliorations significatives pour la supervision de sécurité : nouveaux templates de sécurité officiels, mécanismes d'alerte et d'escalade améliorés, chiffrement renforcé des communications agent-serveur, et intégration SIEM plus facilitée. En 2026, Zabbix n'est plus seulement un outil de monitoring de performance (CPU, RAM, disque, réseau) — c'est une plateforme de supervision de sécurité à part entière, capable de détecter des comportements anormaux, de surveiller l'intégrité des configurations, et d'alerter sur des indicateurs de sécurité précis. Ce guide couvre l'utilisation de Zabbix 7 pour la supervision de sécurité en 2026 : configuration des templates de sécurité Windows et Linux, création de triggers comportementaux (échecs d'authentification répétés, processus suspects, volumes de trafic anormaux), intégration avec les SIEM (Wazuh, Elastic Security), chiffrement PSK et TLS des agents, haute disponibilité (HA) active-passive, et comparaison avec Prometheus/Grafana pour les cas d'usage sécurité. Ces configurations s'appliquent aux déploiements on-premise classiques et aux environnements hybrides (supervision des VMs cloud et des services SaaS via Zabbix proxies).

À RETENIR

À retenir — Zabbix 7 Supervision Sécurité 2026

• **Templates sécurité** : Windows Security, Linux by Zabbix Agent 2, SSL Certificate Check natifs dans Zabbix 7

• **Triggers comportementaux** : seuils sur les Event Logs (EventID 4625), processus inconnus, connexions réseau suspectes

• **Chiffrement obligatoire** : PSK (Pre-Shared Key) ou certificats TLS pour toutes les communications agent-serveur

• **HA Zabbix 7** : cluster active-passive natif sans Pacemaker — failover automatique en quelques secondes

• Intégration SIEM : forwarding des alertes Zabbix vers Wazuh/Elastic via webhook ou script externe

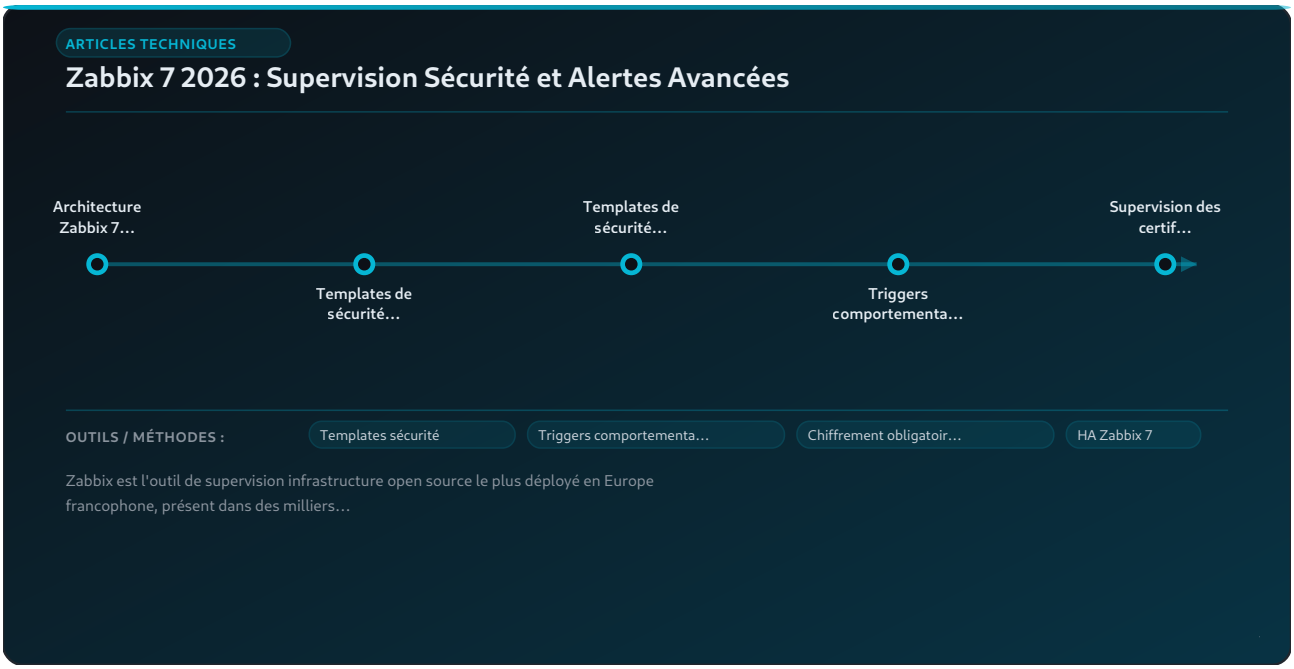
• Zabbix vs Prometheus : Zabbix pour l'infrastructure sécurité "à l'ancienne" (agents, SNMP), Prometheus pour les environnements cloud-native (Kubernetes, microservices)

• **Supervision email sécurité** : vérification SPF/DKIM/DMARC et réputation IP des serveurs de messagerie via scripts Zabbix externes

• **Intégration scanners de vulnérabilités** : Nessus, OpenVAS, Windows Update Agent — suivi continu et alertes sur le retard de patching

• Dashboards automatisés : rapports PDF hebdomadaires Zabbix pour RSSI, analystes SOC et direction

• **Conformité NIS 2 et ISO 27001** : Zabbix documente la surveillance continue avec des métriques exportables pour les audits réglementaires et les contrôles internes



Architecture Zabbix 7 : composants et déploiement

L'architecture Zabbix 7 se compose de plusieurs éléments. Le **Zabbix Server** : le composant central qui collecte les données, évalue les triggers, génère les alertes et stocke les données en base. Le **Zabbix Database** : MySQL/MariaDB ou PostgreSQL (recommandé pour les grandes installations) pour le stockage des métriques et de la configuration. Le **Zabbix Frontend** : l'interface web PHP (Apache ou Nginx) pour la configuration et la visualisation. Les **Zabbix Agents** (Agent ou Agent 2) : installés sur les hôtes supervisés pour la collecte des métriques locales. Le **Zabbix Proxy** : collecteur intermédiaire pour les sites distants ou les réseaux segmentés (idéal pour superviser des sites OT sans ouvrir le réseau industriel vers le Zabbix Server). En 2026, **Zabbix Agent 2** (écrit en Go, contrairement à l'Agent classique en C) est le standard recommandé : support de plus de plugins natifs, meilleure performance, support natif des checks HTTP/HTTPS sans dépendance externe. L'architecture de déploiement recommandée pour un environnement de 500 à 2000 hôtes utilise un serveur dédié Zabbix (8 vCPU / 16 Go RAM) avec PostgreSQL sur un serveur séparé (8 vCPU / 32 Go RAM / SSD NVMe) et des proxies Zabbix dans les segments réseau isolés.

Templates de sécurité Zabbix 7 : Windows Security Monitoring

Zabbix 7 inclut des **templates de sécurité officiels** dans sa bibliothèque de templates, disponibles via l'interface ou sur le portail Zabbix Share. Le template **Windows by Zabbix Agent** inclut des items de sécurité : monitoring des services critiques Windows (Windows Defender, Windows Firewall, Windows Update), vérification de l'activation du pare-feu Windows, état du service de mise à jour automatique. Pour une couverture sécurité plus profonde sur Windows, le template doit être enrichi avec des **items personnalisés** collectant les Event Logs Windows via Zabbix. La collecte d'EventIDs critiques via Zabbix : l'item `eventlog[Security,,,,,]` dans Zabbix permet de collecter les événements du journal de sécurité Windows. Des items plus ciblés peuvent filtrer sur des EventIDs spécifiques : `eventlog[Security,,,"4625",,,]` pour les échecs d'authentification. La **clé Zabbix perf_counter** permet de collecter des compteurs de performance Windows liés à la sécurité (nombre de connexions actives, tentatives d'authentification Kerberos, requêtes LDAP). Les **templates communautaires** disponibles sur GitHub enrichissent encore la couverture : templates pour Windows Defender ATP (métriques de l'antivirus), templates pour les journaux Active Directory (Event IDs Kerberos et NTLM), templates pour IIS (tentatives d'injection dans les logs IIS).

> ⚡ **Retour terrain**
>
> Dans les projets techniques complexes, j'ai appris à toujours commencer par auditer la documentation existante plutôt que l'infrastructure elle-même. Dans 80 % des cas, le delta entre la documentation et la réalité est la source première de risques. Une infrastructure bien documentée qui ne correspond pas à la réalité est plus dangereuse qu'une infrastructure sans documentation — parce qu'elle induit une fausse confiance.

Templates de sécurité Zabbix 7 : Linux Security Monitoring

Pour les hôtes Linux, le template **Linux by Zabbix Agent 2** de Zabbix 7 couvre les métriques système de base. La supervision de sécurité Linux avec Zabbix nécessite des items supplémentaires. La **surveillance du fichier /var/log/auth.log** (ou /var/log/secure sur RHEL/CentOS) via l'item `log[/var/log/auth.log,,,,]` permet de détecter les connexions SSH suspectes, les sudo non autorisés, et les su failures. Le **monitoring des processus actifs** via l'item `proc.num[processname]` alerte sur l'apparition de processus inconnus ou l'arrêt de processus critiques (sshd, firewalld, auditd). La **surveillance des connexions réseau actives** via l'item `net.tcp.listen[port]` détecte l'ouverture de nouveaux ports d'écoute non autorisés — un indicateur d'une backdoor ou d'un service malveillant démarré. Le **monitoring AIDE ou Tripwire** (outils de vérification d'intégrité de fichiers Linux) via Zabbix : un script externe exécuté périodiquement qui retourne le nombre de fichiers modifiés depuis la baseline, déclenche une alerte Zabbix si > 0 pour les fichiers critiques système. La surveillance de l'**état des services de sécurité** (firewalld, SELinux/AppArmor, fail2ban) via Zabbix assure leur disponibilité continue.

Triggers comportementaux : détection d'anomalies avec Zabbix

Les **triggers Zabbix** sont des expressions logiques qui évaluent les données collectées pour détecter des conditions anormales et générer des alertes. Pour la supervision de sécurité, des triggers comportementaux permettent de dépasser le simple seuil de performance pour détecter des anomalies. Le **trigger brute force SSH** : `{Linux_host:log[/var/log/auth.log].count(#10,5m,"regexp","Failed password")}>20` — alerte si plus de 20 échecs de connexion SSH en 5 minutes sur une fenêtre de 10 derniers enregistrements. Le **trigger brute force Windows** basé sur EventID 4625 : alerte si plus de 10 échecs d'authentification Windows en 2 minutes depuis la même source IP. Le trigger **nouveau processus inconnu** : alerte si un processus dont le nom n'est pas dans la liste blanche des processus autorisés est détecté en cours d'exécution — nécessite une liste blanche maintenue à jour par application. Le trigger **connexion sortante suspecte** : alerte si une connexion réseau sortante est établie vers une IP dans une plage

géographique inhabituelle ou vers un port non standard (nécessite un script externe avec géolocalisation). Les **triggers dépendants** dans Zabbix 7 permettent de créer des corrélations : un trigger principal de haute criticité qui désactive des sous-triggers de plus faible criticité pour éviter les cascades d'alertes lors d'une coupure réseau.

Supervision des certificats SSL/TLS avec Zabbix

La **supervision des certificats SSL/TLS** est l'un des cas d'usage sécurité les plus pratiques de Zabbix, évitant les incidents de production liés à l'expiration de certificats. Zabbix 7 inclut un template officiel **SSL Certificate Check** qui vérifie pour chaque domaine supervisé : la date d'expiration du certificat (avec alertes à J-30, J-14, J-7), la validité de la chaîne de certification, le hostname correspondant au certificat, et le protocole TLS supporté (détecter les serveurs utilisant encore TLS 1.0/1.1 non sécurisé). Le **check Zabbix web monitoring** peut être configuré pour simuler des requêtes HTTPS et alerter sur des erreurs SSL (certificat expiré, certificat auto-signé, cipher suite faible). Pour les environnements avec de nombreux certificats internes (PKI d'entreprise), des scripts Zabbix externes peuvent interroger les CA internes pour lister les certificats proches de l'expiration. La **supervision DANE/TLSA** (DNS-based Authentication of Named Entities) pour les domaines email est également possible via des scripts Zabbix, vérifiant l'alignement entre les certificats TLS et les enregistrements DNS TLSA — pertinent pour les organisations avec des exigences de sécurité email avancées.

Chiffrement Zabbix : PSK et TLS pour les agents

Le **chiffrement des communications** entre les agents Zabbix et le serveur est une mesure de sécurité fondamentale, souvent négligée dans les déploiements par défaut. Sans chiffrement, les métriques collectées et les commandes envoyées aux agents transitent en clair sur le réseau — une interception permettrait de connaître la configuration précise de l'infrastructure. Zabbix 7 supporte deux mécanismes de chiffrement. Le **PSK (Pre-Shared Key)** : une clé partagée

configurée à la fois sur le serveur Zabbix et sur chaque agent. La configuration PSK dans l'agent (`zabbix_agent2.conf`) :

`TLSConnect=psk`

`TLSAccept=psk`

`TLSPSKIdentity=psk001`

`TLSPSKFile=/etc/zabbix/zabbix_agentd.psk`

La clé PSK est une chaîne hexadécimale aléatoire générée avec `openssl rand -hex 32`. Le **TLS avec certificats X.509** : plus sécurisé que PSK (support de la révocation via CRL/OCSP, identité plus robuste), mais plus complexe à déployer — nécessite une PKI interne ou des certificats signés. Recommandé pour les environnements avec des exigences de sécurité élevées (DORA, NIS 2). Dans les deux cas, la **politique de chiffrement** doit être configurée en `required` (pas `optional`) pour les agents des segments réseau critiques — interdire les connexions non chiffrées depuis ces zones.

Haute disponibilité Zabbix 7 : cluster natif

Zabbix 7 introduit un **cluster HA natif** (sans Pacemaker ni Corosync) — une fonctionnalité très attendue par les administrateurs Zabbix. La configuration HA Zabbix 7 est simple : dans `zabbix_server.conf`, la directive `HANodeName=node1` (ou node2, node3...) identifie chaque nœud du cluster. Les nœuds partagent la même base de données (haute disponibilité de la base séparée, via PostgreSQL HA avec Patroni par exemple). Le nœud actif est élu via une table de la base de données Zabbix — pas de réseau privé Heartbeat requis. En cas de défaillance du nœud actif, le nœud passif prend le relais en quelques secondes après avoir détecté l'absence d'heartbeat dans la base. Le **frontend Zabbix** doit être configuré pour basculer automatiquement sur le nouveau nœud actif, via un **load balancer** (HAProxy, Nginx upstream) ou une IP virtuelle (Keepalived) pointant vers le nœud actif. La haute disponibilité de la **base de données PostgreSQL** est le composant HA le plus critique — Patroni (HA PostgreSQL automatisé) est la solution recommandée pour les environnements exigeants. Pour un SOC ou une supervision de sécurité critique, le SLA de supervision (Zabbix lui-même doit être surveillé) impose une architecture HA — un monitoring qui tombe est une fenêtre d'attaque non détectée.

Intégration Zabbix avec les SIEM : Wazuh et Elastic

Zabbix est un outil de supervision, pas un SIEM — l'intégration des alertes Zabbix dans un SIEM apporte une corrélation avec les événements de sécurité endpoint et réseau. L'**intégration Zabbix → Wazuh** peut se faire de plusieurs manières. Via les **webhooks Zabbix** : configurer un média webhook dans Zabbix qui envoie les alertes au format JSON vers l'API Wazuh ou vers un script d'ingestion. Via le **forwarding syslog** : configurer un script d'action Zabbix pour écrire les alertes dans syslog, collecté par l'agent Wazuh du serveur Zabbix. Via les **Active Checks Wazuh** : l'agent Wazuh sur le serveur Zabbix peut monitorer le fichier de log d'alertes Zabbix (`/var/log/zabbix/zabbix_server.log`) et envoyer les alertes au Manager Wazuh. L'**intégration Zabbix → Elastic SIEM** est similaire : le webhook Zabbix envoie les alertes vers Logstash (avec un filtre de transformation JSON) qui les index dans Elasticsearch pour analyse dans Kibana Security. Cette intégration permet de corréler une alerte Zabbix "serveur web hors ligne" avec une alerte Wazuh "tentatives de connexion SSH massives sur ce même serveur" — une corrélation qui révèle une attaque DDoS doublée d'un brute force.

Zabbix et supervision des équipements réseau : SNMP et SSH

Zabbix excelle dans la supervision des **équipements réseau** (switches, routeurs, firewalls) via SNMP v2c ou SNMP v3. La supervision SNMP v3 (authentification MD5/SHA + chiffrement DES/AES) est recommandée pour sécuriser les communications avec les équipements réseau — SNMP v1/v2c envoient le community string en clair. Les **templates réseau Zabbix** couvrent les principaux constructeurs (Cisco IOS, Cisco Catalyst, Juniper JunOS, Fortinet FortiOS, Palo Alto, pfSense/OPNsense via SSH checks) avec des métriques de sécurité : état des interfaces (alerte si interface qui devrait être down passe up), volume de trafic par interface (alerte sur des volumes anormaux), état du firewall et des règles ACL, connexions VPN actives, certificats VPN (expiration), CPU/RAM du firewall (alerte sur surcharge qui pourrait indiquer un DDoS). La **supervision des logs firewall via Zabbix** (collecte de syslog réseau) permet des checks sur des patterns spécifiques (nombre de règles DROP par heure, connexions refusées vers des ports

critiques). Pour les équipements sans SNMP, les **checks SSH Zabbix** (exécution de commandes sur l'équipement via SSH et parsing de la sortie) permettent une supervision sur mesure.

Supervision de la sécurité Active Directory avec Zabbix

La **supervision de l'Active Directory** via Zabbix permet de détecter des changements de configuration AD et des anomalies qui peuvent indiquer une compromission. Les items Zabbix pour la supervision AD incluent : la **réplication AD** (temps de réplication entre contrôleurs de domaine, nombre d'objets répliqués — une réplication bloquée ou anormalement rapide peut indiquer un DCSync) via des scripts PowerShell exécutés par l'agent Zabbix. L'**état des contrôleurs de domaine** (réplication, services SYSVOL, NETLOGON, NTDS, Kerberos) via les compteurs de performance Windows et les Event Logs. Le **nombre de comptes lockedout** (EventID 4740) : un pic de comptes verrouillés peut indiquer une attaque de password spraying ou un problème de synchronisation de password. La **taille des groupes sensibles** (Domain Admins, Enterprise Admins, Schema Admins) : un trigger Zabbix alerte si le nombre de membres d'un groupe admin change — ajout non autorisé d'un compte dans Domain Admins est un indicateur critique de compromission. Le **monitoring Kerberos** : nombre de tickets Kerberos générés par heure (un pic anormal peut indiquer un Kerberoasting ou une AS-REP Roasting attack). Ces supervisions AD via Zabbix complètent la détection Wazuh/SIEM avec une dimension d'alerte infrastructure continue.

Zabbix et supervision des sauvegardes

La **supervision des sauvegardes** est souvent sous-estimée mais critique — découvrir qu'une sauvegarde a échoué lors d'un incident de ransomware est une catastrophe. Zabbix peut superviser les jobs de sauvegarde de plusieurs manières. Pour **Proxmox Backup Server (PBS)** : un script Zabbix interroge l'API PBS pour vérifier le statut du dernier job de sauvegarde de chaque VM (succès/échec/date du dernier job) et génère une alerte si la dernière sauvegarde date

de plus de X heures. Pour les outils de sauvegarde Restic/Borg : un script wrapper Restic/Borg exécuté par cron écrit le résultat (succès/échec/taille) dans un fichier de statut, collecté par Zabbix via l'agent. Pour **Veeam Backup** (Windows) : le template Zabbix Veeam disponible en communauté supervise les jobs Veeam via PowerShell et les API Veeam. Pour les sauvegardes cloud (AWS S3, Azure Backup) : des scripts Zabbix externe interrogent les APIs cloud pour vérifier l'état des dernières sauvegardes. Un trigger Zabbix "**backup failed**" doit avoir la plus haute criticité (Disaster) — une sauvegarde manquée est toujours urgente, même si ce n'est pas encore un incident de sécurité. La supervision Zabbix des sauvegardes crée une alerte proactive avant que le problème ne devienne critique, contrairement à la découverte réactive lors d'une tentative de restauration.

Zabbix vs Prometheus/Grafana : quel outil pour la sécurité ?

Le débat **Zabbix vs Prometheus** est récurrent dans les équipes infrastructure et sécurité. En 2026, les deux outils ont des cas d'usage différents et complémentaires. **Zabbix** excelle pour : la supervision d'agents sur des endpoints Windows/Linux classiques, la supervision SNMP d'équipements réseau, les vérifications actives (TCP checks, HTTP checks, ICMP), les environnements mixtes avec des équipements legacy, les alertes complexes avec dépendances et escalades. **Prometheus** excelle pour : les environnements cloud-native (Kubernetes, conteneurs Docker), les métriques exposées en format Prometheus (99% des applications modernes), la scalabilité horizontale, l'intégration avec Grafana pour des dashboards avancés. Pour un SOC sécurité, **Zabbix est généralement le choix principal** pour la supervision infrastructure sécurité (disponibilité des services de sécurité, agents endpoint, équipements réseau, sauvegardes) car ses agents et son modèle push/pull sont plus adaptés aux environnements hétérogènes. Prometheus/Grafana complète Zabbix pour les métriques applicatives et cloud-native. Mon opinion : dans la plupart des PME et ETI françaises avec des infrastructures Windows/VMware/physique, Zabbix est plus adapté que Prometheus — sa courbe d'apprentissage est plus abrupte initialement mais sa puissance de supervision des environnements traditionnels n'a pas d'équivalent open source.

Alertes et escalades dans Zabbix 7 : configuration avancée

La gestion des **alertes et escalades** dans Zabbix 7 est un mécanisme puissant pour s'assurer que les incidents de sécurité sont traités dans les délais appropriés. Le mécanisme d'**escalade Zabbix** : si une alerte de sécurité n'est pas acquittée dans N minutes, elle est automatiquement escaladée au niveau supérieur (L1 → L2 → Responsable sécurité → RSSI → DG). La configuration des **médias d'alerte** dans Zabbix 7 inclut : Email (SMTP avec TLS), SMS (via passerelles HTTP), Slack (webhook Zabbix natif pour Slack), Microsoft Teams (webhook), PagerDuty (intégration native), Telegram (bot Telegram via script externe), et Webhook générique (pour n'importe quelle API). Les **actions Zabbix** permettent de définir des règles complexes : alerter par email uniquement en heures ouvrées, passer à SMS hors-heures, escalader à J+30 minutes si pas d'acquittement. L'**acquittement des alertes** dans l'interface Zabbix permet aux analystes de marquer une alerte comme prise en compte, avec un commentaire, directement depuis l'interface web ou l'application mobile Zabbix. Les **tags Zabbix 7** (étiquettes sur les hôtes, items et triggers) permettent de filtrer et router les alertes : un tag "security" sur tous les triggers de sécurité permet de les afficher dans un dashboard dédié et de les router vers le canal Slack SOC plutôt que le canal générique infrastructure.

Zabbix et conformité : NIS 2, ISO 27001 et PCI-DSS

La supervision Zabbix peut contribuer directement à la **conformité réglementaire**. Pour **NIS 2** (directive applicable aux entités essentielles et importantes françaises) : Zabbix peut documenter la surveillance continue des systèmes (exigence A.12 de la norme), les alertes sur les disponibilités des services critiques, et les métriques de performance des mesures de sécurité (taux de disponibilité des services critiques). Les rapports Zabbix exportés en PDF peuvent servir de preuves pour les audits NIS 2 réalisés par l'ANSSI ou des auditeurs tiers. Pour **ISO 27001** (annex A contrôle A.12.1 "Operational procedures and responsibilities" et A.12.4 "Logging and monitoring") : la configuration Zabbix (quoi superviser, qui est alerté, délais d'escalade) doit être documentée dans les procédures d'exploitation de l'ISMS. Pour **PCI-DSS v4.0** (applicable aux organisations traitant des données de paiement) : l'exigence 10.6 impose une revue quotidienne

des logs — Zabbix peut automatiser la détection des événements significatifs dans les logs des systèmes PCI-DSS scope, réduisant la charge de la revue manuelle. La génération de **rapports de supervision Zabbix** (dashboards, historiques d'alertes, disponibilité des services) est facilement exportable pour les auditeurs — un avantage pratique souvent cité par les RSSI utilisant Zabbix.

Performance Zabbix : tuning pour les grands environnements

Pour les **grands environnements Zabbix** (1000+ hôtes, millions de métriques par heure), le tuning des performances est essentiel pour maintenir la réactivité du système et éviter les alertes manquées dues à la saturation. Les **paramètres de performance Zabbix Server** clés : `StartPollers` (nombre de threads de collecte active — augmenter si la file d'attente des pollers est > 0 en permanence), `StartTrappers` (threads pour les données passive/push — augmenter pour les environnements avec beaucoup d'agents en mode push), `CacheSize` (taille du cache de configuration en RAM — augmenter pour les grandes configurations). Le **tuning PostgreSQL** est souvent la clé des performances Zabbix : `shared_buffers` (25% de la RAM), `work_mem`, `effective_cache_size`, `checkpoint_completion_target`. L'utilisation de **TimescaleDB** (extension PostgreSQL pour les séries temporelles) est officiellement supportée par Zabbix 7 et offre des performances de requête historique 10x meilleures sur les grands ensembles de données — idéal pour les analyses de tendances de sécurité sur des mois. La **politique d'housekeeper Zabbix** (suppression des données anciennes selon les politiques de rétention configurées par item) doit être planifiée pour les heures creuses — une housekeeper qui tourne en plein jour peut dégrader les performances de la supervision active et retarder la détection d'alertes de sécurité critiques pendant l'exécution des opérations de nettoyage.

Zabbix API : automatisation et intégration avec les outils DevSecOps

L'**API JSON-RPC de Zabbix** est une interface programmatique complète permettant d'automatiser toutes les opérations de configuration et de collecte. L'API Zabbix permet de : créer des hôtes automatiquement lors du provisionnement d'une nouvelle VM (intégration avec Ansible, Terraform, scripts de déploiement), lire les dernières valeurs des métriques pour des tableaux de bord externes, récupérer les alertes actives et leur statut pour les dashboards de management, et modifier la configuration Zabbix en masse (mise à jour des templates sur des centaines d'hôtes). Pour l'**intégration DevSecOps** : dans un pipeline CI/CD, après le déploiement d'une application, l'API Zabbix peut automatiquement ajouter le nouveau serveur à la supervision avec les templates appropriés — infrastructure-as-code appliqué au monitoring. Des **outils Terraform pour Zabbix** (provider `terraform-provider-zabbix`) permettent de définir la configuration Zabbix en code Terraform — hôtes, templates, triggers, groupes d'hôtes gérés en Git avec les mêmes workflows DevOps que l'infrastructure. L'API Zabbix est également utilisée par **Shuffle** pour des intégrations SOAR : un workflow Shuffle peut récupérer les métriques Zabbix d'un hôte en cours d'investigation pour enrichir un case TheHive avec le contexte de supervision de l'équipement suspect.

Zabbix Proxy pour les réseaux segmentés et les sites distants

Le **Zabbix Proxy** est un composant essentiel pour les architectures réseau avec des segments isolés (OT/ICS, DMZ, sites distants) ou des contraintes de firewall qui empêchent les agents de communiquer directement avec le Zabbix Server. Le Proxy collecte les données localement et les transmet au Server de manière asynchrone — si la liaison réseau est temporairement interrompue, les métriques sont bufferisées dans le Proxy et transmises dès que la connexion est rétablie. Pour les **réseaux OT/ICS**, le Zabbix Proxy est déployé dans la DMZ industrielle et collecte les métriques des automates (PLC) via SNMP ou via des items dédiés Modbus/DNP3 (avec des scripts externes) sans que le Zabbix Server en réseau IT n'ait un accès direct au réseau OT. Le Proxy peut être configuré en mode **actif** (il initie la connexion vers le Server — plus adapté aux environments avec des firewalls stateful) ou **passif** (le Server interroge le Proxy). Le

chiffrement entre Proxy et Server utilise les mêmes mécanismes PSK ou TLS que pour les agents. Dans un déploiement multi-sites, chaque site dispose de son Proxy Zabbix local, réduisant la dépendance à la liaison WAN pour la supervision locale tout en centralisant les données sur le Zabbix Server principal.

FAQ — Zabbix 7 Supervision Sécurité 2026

Zabbix peut-il remplacer un SIEM pour la supervision de sécurité ?

Non, Zabbix ne remplace pas un SIEM — les deux sont complémentaires. Zabbix excelle pour la supervision continue de disponibilité et de performance avec des alertes infrastructure (serveur down, certificat expirant, processus arrêté). Un SIEM comme Wazuh ou Elastic Security excelle pour la corrélation d'événements de sécurité (logs d'authentification, détection de malwares, analyse comportementale) et l'investigation forensique. L'architecture idéale combine les deux : Zabbix pour le monitoring infrastructure et les alertes opérationnelles, le SIEM pour la détection des menaces et la réponse à incident. Les alertes Zabbix peuvent être forwardées vers le SIEM pour enrichir la corrélation globale.

Quelle est la fréquence de collecte recommandée pour les métriques de sécurité dans Zabbix ?

Les métriques de sécurité critiques (échecs d'authentification, processus actifs, connexions réseau) doivent être collectées toutes les 30 à 60 secondes pour permettre une détection rapide. Les vérifications de certificats SSL peuvent être exécutées toutes les heures. Les vérifications de conformité (état des services de sécurité, configuration firewall) toutes les 5 à 15 minutes. Les métriques de performance (CPU, RAM, disque) toutes les 60 secondes en standard, 15 secondes pour les composants critiques. Augmenter la fréquence de collecte a un coût en performance Zabbix — trouver l'équilibre entre réactivité des alertes de sécurité et charge du serveur Zabbix.

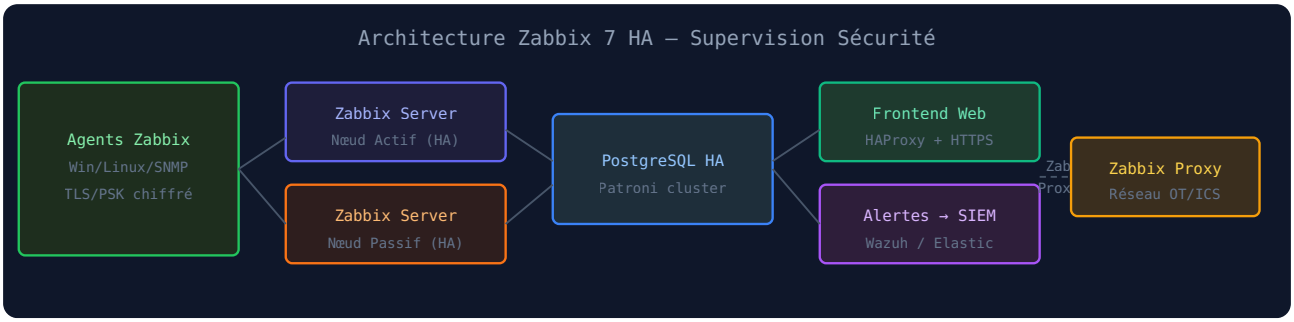
Comment sécuriser l'interface web Zabbix elle-même contre les attaques ?

L'interface web Zabbix est une cible d'attaque — sa compromission donne accès à la configuration de toute l'infrastructure supervisée. Mesures recommandées : placer le frontend Zabbix derrière un reverse proxy Nginx avec restriction IP (accès uniquement depuis le réseau de management ou VPN), activer la double authentification Zabbix (TOTP depuis Zabbix 7), configurer HTTPS strict (TLS 1.3 uniquement, HSTS), désactiver les comptes par défaut (Admin Zabbix par défaut doit être renommé ou supprimé), limiter les permissions des comptes via les rôles Zabbix 7 (nouveaux rôles personnalisables), et monitorer les logs d'accès Nginx du frontend Zabbix pour les tentatives de force brute sur la page de login.

Zabbix 7 supporte-t-il la supervision de Kubernetes et des environnements conteneurisés ?

Oui, Zabbix 7 supporte Kubernetes via le template officiel Kubernetes by HTTP qui supervise les métriques des nœuds, pods, namespaces et ressources via l'API Kubernetes. Pour les conteneurs Docker, le template Docker by Zabbix Agent 2 supervise l'état des conteneurs, leur consommation de ressources, et les logs. Cependant, pour les environnements cloud-native modernes avec des centaines de pods dynamiques, Prometheus/Grafana reste plus adapté — Zabbix est meilleur pour les environnements avec des hôtes relativement stables. Une architecture hybride Zabbix (pour l'infrastructure Kubernetes physique et les worker nodes) + Prometheus (pour les métriques applicatives des pods) est souvent adoptée.

Fonctionnalité	Zabbix 7	Prometheus+Grafana	Datadog/New Relic
Agents endpoints	Natifs (Win/Linux)	Node Exporter	Agents natifs
SNMP équipements réseau	Excellent (v1/v2/v3)	SNMP Exporter	Supporté
Kubernetes / cloud-native	Supporté (limité)	Natif et excellent	Excellent
HA intégré	Natif (v7.0)	Via Thanos/Cortex	SaaS (always-on)
Coût licence	Gratuit	Gratuit	15-25$/hôte/mois



Zabbix et supervision de la sécurité email : SPF, DKIM, DMARC

La sécurité email est un vecteur d'attaque majeur — phishing, BEC (Business Email Compromise), et usurpation de domaine. Zabbix peut superviser en continu les **mécanismes de sécurité email** de l'organisation. La **vérification des enregistrements SPF** via un item Zabbix externe (script Python ou Perl utilisant dnspython) : alerte si l'enregistrement SPF du domaine change (ajout non autorisé d'un serveur autorisé à envoyer des emails) ou est invalide. La **vérification DMARC** : contrôle que l'enregistrement DMARC (_dmarc.domaine.fr) existe, est valide, et est en mode `p=reject` ou `p=quarantine` (pas `p=none` qui ne bloque rien). La supervision des **rapports DMARC** (fichiers XML envoyés par les serveurs de réception) est plus complexe mais permet de détecter des tentatives d'usurpation du domaine — un service externe comme dmarcian ou parsedmarc peut être interrogé via API par Zabbix. La **surveillance de la réputation email** des IP des serveurs de messagerie de l'organisation : des scripts Zabbix peuvent interroger des RBL (Real-time Blackhole List — Spamhaus, SORBS, Barracuda) pour détecter si une IP de l'organisation a été blacklistée (indicateur d'un compte email compromis envoyant du spam). Ces supervisions email sont souvent absentes des déploiements Zabbix orientés infrastructure mais apportent une valeur sécurité significative pour les organisations qui envoient des emails critiques (factures, communications clients, notifications administratives).

Supervision de la disponibilité des services de sécurité

Zabbix est l'outil idéal pour superviser la **disponibilité des services de sécurité eux-mêmes** — s'assurer que les outils de protection fonctionnent en permanence. Les services de sécurité à superviser via Zabbix incluent : **Antivirus/EDR** — vérification que l'agent de protection endpoint est actif sur chaque serveur et poste (via un item Zabbix vérifiant le processus ou le service Windows/Linux correspondant) ; **Firewall** — vérification de la disponibilité des firewalls physiques et virtuels (ping, SNMP health check) ; **VPN concentrateur** — disponibilité et nombre de connexions VPN actives (alerte si le concentrateur VPN tombe hors-heures alors que des employés en télétravail y sont connectés) ; **SIEM/Wazuh** — disponibilité du Manager Wazuh et de l'indexeur (alerte si le SIEM lui-même est down — une fenêtre de non-détection critique) ; **Solution de sauvegarde** — état des jobs de sauvegarde PBS/Veeam/Restic ; **Meilisearch/LDAP/RADIUS** — services d'authentification et d'annuaire dont la défaillance impacte tous les accès. La **supervision circulaire** (Zabbix qui surveille Zabbix) est gérée par une instance Zabbix secondaire ou par des checks depuis l'extérieur — la surveillance du superviseur est une exigence souvent oubliée mais fondamentale pour garantir l'absence d'angle mort dans la détection.

Supervision de la surface d'attaque externe avec Zabbix

Zabbix peut être utilisé pour une **supervision de la surface d'attaque externe** — surveiller en continu les services exposés sur Internet pour détecter les changements non autorisés. Les **checks HTTP/HTTPS de Zabbix** permettent de vérifier la disponibilité et le contenu des pages web de l'organisation (alerte si une page web est défigurée — le contenu diffère de la référence connue). La **vérification des ports ouverts** sur les IP publiques de l'organisation via Zabbix (checks TCP sur les ports standards et non standards) détecte l'ouverture intempestive d'un service non autorisé exposé sur Internet. Le **monitoring DNS** via Zabbix (items `net.dns`) surveille les enregistrements DNS de l'organisation — alerte si les serveurs de noms, les enregistrements SPF/DKIM/DMARC ou les enregistrements A de services critiques changent (indicateur de DNS hijacking ou de modification non autorisée). La **supervision des certificats wildcard** (souvent

utilisés pour *.example.fr) avec alertes d'expiration est particulièrement critique — un certificat wildcard expiré impacte tous les sous-domaines simultanément. Des **scénarios web Zabbix** (suites de requêtes HTTP simulant une navigation utilisateur) peuvent vérifier le bon fonctionnement des flux d'authentification critiques (login page, portail VPN, webmail) et alerter si un service d'authentification est down ou renvoie une erreur inhabituelle.

Zabbix et gestion des vulnérabilités : intégration avec les scanners

La **gestion des vulnérabilités** est un processus clé de la sécurité qui s'intègre naturellement avec Zabbix. Si Wazuh Agent 2 inclut une fonction d'évaluation des vulnérabilités basée sur les packages installés, Zabbix peut compléter cette approche avec des intégrations de scanners de vulnérabilités dédiés. L'**intégration Nessus/Tenable avec Zabbix** : des scripts Zabbix externes interrogent l'API Tenable Security Center ou Tenable.io pour récupérer le nombre de vulnérabilités critiques/hautes par hôte, et créent des triggers Zabbix si un hôte critique présente des vulnérabilités non corrigées depuis plus de X jours. L'**intégration OpenVAS/Greenbone** (scanner de vulnérabilités open source) avec Zabbix suit le même principe — l'API OpenVAS peut être interrogée pour les résultats de scan, et Zabbix alerte sur les nouvelles vulnérabilités critiques découvertes. La **supervision du patch management Windows** via Zabbix : l'item `wua.updates[критerion]` (Windows Update Agent) compte les mises à jour Windows en attente ; un trigger alerte si un serveur Windows présente des mises à jour critiques non installées depuis plus de 7 jours. Ce suivi Zabbix du statut de patching complète un scanner de vulnérabilités en donnant une vue continue de l'état de patching de chaque hôte, sans attendre le prochain scan périodique.

Retour terrain : Zabbix dans une administration française (350 hôtes)

Un **retour d'expérience concret** : une collectivité territoriale française (environ 350 postes, 80 serveurs, 40 équipements réseau) a déployé Zabbix 7 en 2025 pour remplacer un outil de

monitoring propriétaire dont la licence devenait trop coûteuse. Le projet a duré 3 mois : 1 mois de déploiement et configuration initiale (agents sur tous les serveurs, SNMP sur les équipements réseau, templates standards), 2 mois de tuning pour réduire les faux positifs à un niveau acceptable (moins de 20 alertes qualifiées par jour). Les résultats après 6 mois : détection proactive de 3 disques en voie de saturation avant incident, identification d'un serveur dont les sauvegardes échouaient silencieusement depuis 3 semaines, et alerte précoce sur un certificat SSL expirant qui aurait interrompu l'accès au portail citoyen. L'**opinion de l'administrateur système en charge** : la plus grande difficulté n'est pas l'installation de Zabbix mais la définition de ce qu'on veut superviser (inventaire exhaustif des hôtes et services critiques) et le tuning des triggers pour éviter la fatigue des alertes. Trois mois après le déploiement, l'équipe avait réduit les faux positifs de 90% grâce aux exceptions et aux ajustements de seuils. L'interface web de Zabbix 7, modernisée par rapport aux versions précédentes, a été bien accueillie par les administrateurs juniors de l'équipe.

Dashboards de sécurité Zabbix : conception et partage

Les **dashboards Zabbix 7** ont été significativement améliorés — ils supportent désormais des widgets plus riches (graphiques, cartes, top N, carte géographique) et peuvent être partagés facilement avec des parties prenantes non-techniques. Pour la supervision de sécurité, la conception des dashboards suit des audiences différentes. Le **dashboard RSSI** (vue exécutive) : taux de disponibilité des services critiques sur 30 jours (en %), nombre d'incidents de sécurité du mois, statut des certificats SSL (tous verts = aucun expirant dans 30 jours), état de la conformité (hôtes avec mises à jour en retard). Le **dashboard analyste SOC** (vue opérationnelle) : alertes actives triées par criticité, hôtes avec le plus d'alertes des dernières 24 heures, graphiques de tentatives d'authentification (pour détecter visuellement un pic de brute force), carte des hôtes avec indicateur de statut (rouge = alerte critique active). Le **dashboard infrastructure** (vue technique) : état des services Zabbix lui-même (queue de polling, performance du serveur), métriques de collecte (items/seconde, nouvelles valeurs/seconde), état de la base de données PostgreSQL (connexions actives, taille de la DB, performances des requêtes). Les dashboards Zabbix 7 peuvent être **exportés en PDF** automatiquement via le scheduler de rapports intégré —

l'envoi automatique d'un rapport de supervision hebdomadaire par email à la direction est une pratique qui améliore la visibilité de l'équipe sécurité sans nécessiter d'effort manuel supplémentaire.

Macros Zabbix et gestion des seuils par environnement

Les **macros Zabbix** sont des variables réutilisables qui permettent de paramétrer les templates sans modifier les items et triggers directement. Cette fonctionnalité est particulièrement utile pour la supervision de sécurité où les seuils d'alerte varient selon l'environnement ou l'hôte. La macro `{$MAX_AUTH_FAILURES}` peut être définie globalement à 10 (10 échecs d'authentification = alerte), surridée à 50 sur les serveurs de test où des outils de test d'authentification s'exécutent légitimement, et surridée à 3 sur les serveurs de base de données critiques où même 3 échecs doivent alerter immédiatement. Les **macros de niveau hôte** permettent une personnalisation fine sans dupliquer les templates. La macro `{$CERT_EXPIRE_WARN}` définit le seuil d'alerte pour les certificats SSL : 30 jours par défaut, 60 jours pour les certificats wildcard (renouvellement plus complexe nécessitant plus d'anticipation). Les **macros avec contexte** (introduites dans Zabbix 5.4, disponibles en 7.0) permettent des seuils différents selon le contexte d'application de la macro — une flexibilité avancée pour les grandes configurations avec des centaines d'hôtes de types différents. La **documentation des macros** (chaque macro avec sa description et sa valeur par défaut) est une pratique de configuration recommandée pour faciliter la maintenance par des administrateurs qui reprennent une configuration Zabbix existante.

Zabbix 7 représente en 2026 la solution de supervision de sécurité open source la plus polyvalente pour les infrastructures hybrides françaises — sa capacité à combiner la supervision infrastructure classique (SNMP, agents, checks TCP) avec des vérifications de sécurité avancées (Event Logs, scripts externes, API) en fait un outil indispensable dans toute architecture SOC. Son cluster HA natif, son API complète, et son intégration naturelle avec les SIEM open source (Wazuh) et les outils SOAR (Shuffle) l'ancrent définitivement dans l'écosystème des outils de sécurité modernes, au-delà de son rôle historique de simple monitoring infrastructure. Pour une supervision de sécurité complète, Zabbix s'inscrit dans une architecture SOC plus large : consultez notre comparatif des outils SOC open source Wazuh, TheHive, MISP et Shuffle pour la

détection et la réponse aux incidents, notre guide sur le durcissement Proxmox VE 8 pour l'infrastructure à superviser, notre article sur les protocoles OT et leur sécurisation pour les environnements industriels surveillés par Zabbix Proxy, et notre guide sur le pentest OT/ICS pour les tests de sécurité des infrastructures supervisées. Pour les ressources officielles, la documentation Zabbix 7.0 et la bibliothèque de templates Zabbix Share sont les références incontournables de la communauté internationale.

