AYI NEDJIMI Consultants

Gestion des vulnérabilités DevSecOps : triage et remède

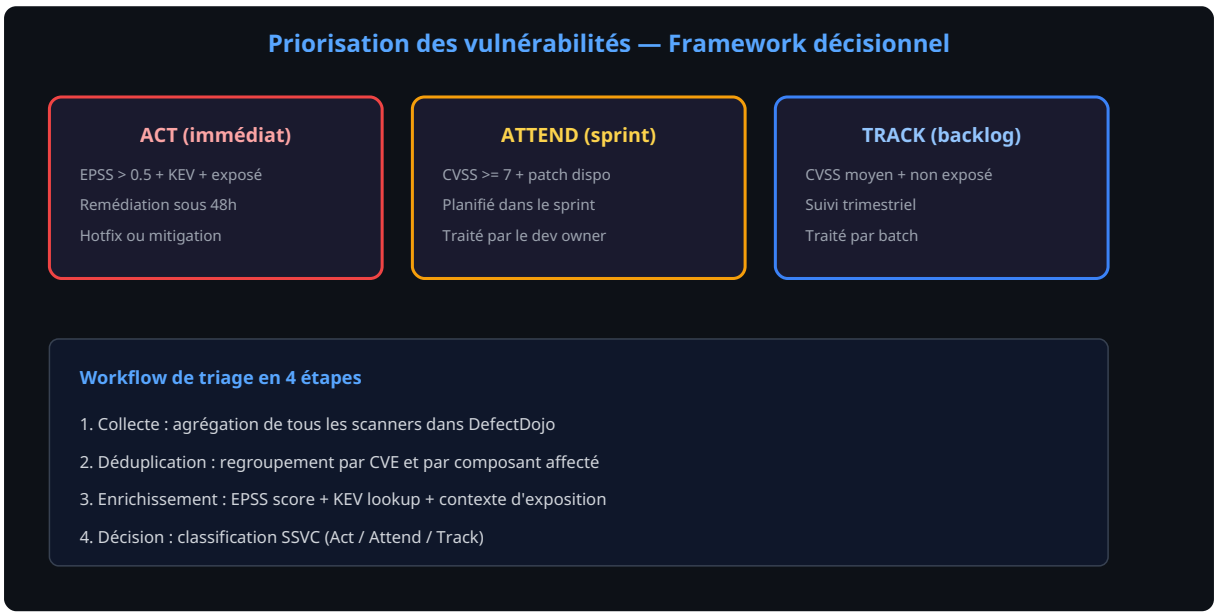
Catégorie : DevSecOps Lecture : 5 min Publié le : 12/03/2026 Auteur : Ayi NEDJIMI

Gestion des vulnérabilités DevSecOps : méthodes de triage, priorisation par EPSS et SSVC, workflows de remédiation et intégration avec vos outils.

Votre pipeline CI/CD remonte 847 vulnérabilités. Votre backlog sécurité déborde. Votre équipe de 5 développeurs ne peut raisonnablement en traiter que 20 par sprint. Par laquelle commencer ? Si votre réponse est "celles avec le score CVSS le plus élevé", vous faites la même erreur que 80% des organisations. Le CVSS mesure la sévérité théorique, pas le risque réel dans votre contexte. Une vulnérabilité CVSS 9.8 dans un composant isolé derrière trois couches de réseau est moins urgente qu'une CVSS 7.0 sur votre API publique exposée à Internet. La gestion des vulnérabilités en DevSecOps exige un framework de priorisation contextuel qui prend en compte l'exploitabilité réelle, l'exposition de l'actif, l'existence de correctifs et l'impact métier. Ce guide vous présente les méthodes modernes de triage — EPSS, SSVC, KEV — et les workflows de remédiation qui fonctionnent dans un contexte agile. Vous découvrirez comment passer de 847 findings ingérables à 25 actions prioritaires par sprint, avec des exemples concrets d'intégration dans vos outils existants.

Points clés à retenir

- **EPSS** (Exploit Prediction Scoring System) prédit la probabilité d'exploitation dans les 30 jours — bien plus actionnable que le CVSS seul
- Le catalogue **KEV** (Known Exploited Vulnerabilities) de la CISA liste les vulnérabilités activement exploitées — traitez-les en priorité absolue
- **SSVC** (Stakeholder-Specific Vulnerability Categorization) de la CISA produit des décisions claires : Track, Track*, Attend, Act
- Le triage contextuel réduit de 90% le volume de vulnérabilités à traiter en urgence sans augmenter le risque réel



Pourquoi le CVSS seul ne suffit pas

Le **CVSS** (Common Vulnerability Scoring System) évalue la sévérité intrinsèque d'une vulnérabilité. C'est une mesure technique, pas une mesure de risque. Un CVSS 9.8 signifie que la vulnérabilité est potentiellement dévastatrice — mais pas qu'elle sera exploitée demain. Sur les 20 000+ CVE publiées chaque année, moins de 5% font l'objet d'un exploit actif dans les 30 jours. Traiter les 20 000 avec la même urgence est impossible et contre-productif.

Le CVSS manque de deux dimensions clés : la probabilité d'exploitation et le contexte de votre infrastructure. C'est là qu'interviennent **EPSS** et **SSVC**. Pour les vulnérabilités détectées par vos scanners, consultez notre comparatif SAST, DAST et IAST qui détaille les types de findings produits par chaque outil.

EPSS : prédire l'exploitation réelle

L'**Exploit Prediction Scoring System** (FIRST.org) utilise du machine learning pour estimer la probabilité qu'une CVE soit exploitée dans les 30 jours suivants. Le score va de 0 à 1. Un EPSS de 0.9 signifie 90% de chances d'exploitation active — traitez-la immédiatement. Un EPSS de 0.01 signifie 1% — planifiable dans le backlog.

En combinant CVSS et EPSS, vous obtenez une priorisation bien plus efficace :

CVSS	EPSS	Décision	Délai
>= 9.0	> 0.5	**ACT** immédiatement	24-48h
>= 7.0	> 0.1	**ATTEND** ce sprint	1-2 semaines
>= 7.0	< 0.1	**TRACK*** surveillance	Prochain trimestre
< 7.0	< 0.1	**TRACK** backlog	Best effort

Cette matrice réduit typiquement de 80-90% le volume de vulnérabilités à traiter en urgence. Le catalogue KEV de la CISA complète l'EPSS en listant les vulnérabilités avec une exploitation confirmée — toute CVE dans le KEV est automatiquement ACT.

SSVC : un framework de décision structuré

Le **Stakeholder-Specific Vulnerability Categorization** (SSVC), développé par la CISA et Carnegie Mellon, va au-delà du scoring pour produire une décision. L'arbre de décision SSVC intègre quatre facteurs : l'état d'exploitation (active, PoC, aucune), l'automatisabilité de l'attaque, l'impact technique et l'impact sur la mission. La sortie est une action : Track, Track*, Attend ou Act.

Concrètement, SSVC transforme une discussion subjective ("c'est critique ou pas ?") en un processus reproductible et documenté. Chaque décision est traçable et auditable — un atout pour la conformité NIS2 et ISO 27001 qui exigent un processus de gestion des vulnérabilités documenté.

Workflow de remédiation en contexte agile

Le triage produit des actions. Encore faut-il les intégrer dans votre processus de développement. Voici le workflow qui fonctionne dans un contexte Scrum :

- **ACT** — Hotfix immédiat, hors sprint si nécessaire. Le security champion de l'équipe prend le lead. Notification Slack/Teams automatique depuis **DefectDojo**.
- **ATTEND** — User story de type "bug security" ajoutée au sprint backlog. Traitement par le développeur propriétaire du composant. Définition of Done : vuln fermée dans DefectDojo.
- **TRACK** — Regroupement par batch trimestriel. Un sprint entier par trimestre est consacré au "tech debt security". Montée de version des dépendances, nettoyage des configurations.

L'outil central est **DefectDojo** ou **Dependency-Track** qui agrège les findings de tous vos scanners (Semgrep, Trivy, ZAP, Gitleaks) et applique les règles de priorisation automatiquement. L'intégration bidirectionnelle avec Jira ou Linear permet de créer les tickets directement depuis l'interface. Pour le pipeline qui alimente DefectDojo, consultez notre guide pipeline CI/CD sécurisé.

Métriques de suivi de la remédiation

La gestion des vulnérabilités sans métriques, c'est piloter à l'aveugle. Voici les KPI à suivre :

- **MTTR** (Mean Time To Remediate) par sévérité — Cible : CRITICAL < 48h, HIGH < 7j, MEDIUM < 30j.
- **Vulnérabilités ouvertes par âge** — Le "aging" des vulnérabilités révèle les points de blocage.
- **Taux de réouverture** — Une vulnérabilité corrigée puis réintroduite signale un problème systémique.
- **Couverture de scan** — % des applications et dépôts couverts par au moins un scanner.

- **Ratio ACT/total** — Si plus de 10% de vos findings sont ACT, vos scanners génèrent trop de bruit ou votre infrastructure est trop exposée.

Ces métriques alimentent votre tableau de bord DevSecOps. Le rapport NIST sur la qualité logicielle fournit des benchmarks sectoriels pour comparer vos performances.

Automatiser le triage avec des politiques

Le triage manuel ne passe pas à l'échelle au-delà de 100 findings par semaine. Automatisez les décisions évidentes :

```
# Politique de triage automatique (pseudo-code DefectDojo)
if cve in CISA_KEV:
    severity = "Critical"
    action = "ACT"
    sla_hours = 48
elif epss > 0.5 and cvss >= 7.0:
    severity = "High"
    action = "ATTEND"
    sla_days = 14
elif cvss >= 7.0 and epss < 0.1:
    severity = "Medium"
    action = "TRACK_STAR"
    sla_days = 90
else:
    action = "TRACK"
```

DefectDojo supporte les règles de groupe (grouping rules) et les SLA personnalisés par sévérité. Configurez-les une fois, et chaque nouveau finding est automatiquement classifié, assigné et suivi. Cela libère votre équipe sécurité pour se concentrer sur les cas ambigus et le threat modeling. Pour la détection des secrets qui alimentent ce flux, consultez notre guide sur Gitleaks et TruffleHog.

Sources et références : OWASP DevSecOps · NIST

Questions fréquentes sur la gestion des vulnérabilités

Comment gérer les vulnérabilités sans correctif disponible ?

Deux options : la mitigation (WAF rule, network segmentation, désactivation de la fonctionnalité affectée) ou l'acceptation documentée du risque. Créez un registre des risques acceptés avec une date de revue trimestrielle. Si un correctif est publié ultérieurement, votre outil de monitoring SBOM vous alertera automatiquement.

Faut-il un outil dédié ou Jira suffit-il pour le suivi ?

Jira seul ne suffit pas pour 3 raisons : pas de déduplication native (un même CVE trouvé par 3 scanners = 3 tickets), pas d'enrichissement automatique (EPSS, KEV), et pas de vue consolidée par composant. DefectDojo ou Dependency-Track en amont de Jira est la bonne architecture. Les tickets Jira sont créés uniquement pour les findings ACT et ATTEND.

Quel SLA appliquer par niveau de sévérité ?

Les benchmarks du secteur (données Veracode State of Software Security 2024) sont : CRITICAL 15 jours, HIGH 30 jours, MEDIUM 90 jours. Mes recommandations pour une organisation mature : CRITICAL 48h, HIGH 7 jours, MEDIUM 30 jours. Adaptez ces SLA à votre contexte : une API publique fintech n'a pas les mêmes exigences qu'un outil interne de reporting.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr