



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*



S-SDLC / DEVSECOPS

QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?



Résumé exécutif

Messages clés de l'ANSSI pour l'écosystème numérique

Étude de marché S-SDLC/DevSecOps

1. Introduction
2. Analyse du marché et tendances structurelles
3. Focus sur la dynamique de la demande
4. Focus sur la dynamique de l'offre
5. Alignement offre-demande

Annexes

6. Méthodologie de l'analyse de marché
7. Scénarios d'attaque des pipeline CI/CD
8. Synthèse des réglementations
9. Aperçu de la recherche académique
10. Sélection des catégories stratégiques

Outils

- a. Pipelines CI/CD – Activités & composants clés
- b. Pipelines CI/CD – Analyse de risque et recommandations
- c. Pipelines CI/CD – Feuilles de route de haut niveau
- d. Frameworks SAMM et SLSA



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

RÉSUMÉ EXÉCUTIF



Contexte, objectifs et méthodologie de l'étude

Contexte

Le paysage du **S-SDLC*** et du **DevSecOps** est structuré par des **pressions significatives**, résultant à la fois de **menaces systémiques récurrentes ciblant la chaîne d'approvisionnement** (p. ex. attaques SolarWinds, XZ Utils) et du **renforcement des exigences réglementaires** (p. ex. Cyber Resilience Act - CRA).

Dans le cadre de **la mission de politique industrielle qu'elle porte pour l'ANSSI**, la division **Industrie et Technologie** souhaite renforcer sa **connaissance du marché S-SDLC et DevSecOps** afin d'identifier d'éventuelles lacunes nécessitant une intervention au titre de la politique industrielle.

Objectifs

Les principaux objectifs de cette étude sont de :

- Fournir une vision globale du **marché S-SDLC et DevSecOps européen**
- Suggérer des recommandations concrètes **pour répondre aux lacunes identifiées sur le marché**

Cette étude vise également à fournir des lignes directrices pour **sécuriser les pipelines CI/CD****.

Méthodologie – Approche de recherche et d'analyse



Analyse des données publiques

- **Identification des catégories de solutions de sécurité S-SDLC/DevSecOps.**
- Analyse du **cadre réglementaire** applicable au S-SDLC/DevSecOps.
- Revue de la **littérature académique et professionnelle** liée au S-SDLC/DevSecOps.



Questionnaires Fournisseurs & Organisations

Fournisseurs : Conception d'un questionnaire combinant une analyse quantitative et qualitative des fournisseurs et de leurs solutions.

Organisations : Conception d'un questionnaire structuré basé sur « **OWASP SAMM – modèle de maturité DevSecOps** ».



Entretiens & consolidation des informations

Sur la base des résultats des questionnaires, **des entretiens** ont été réalisés avec **les fournisseurs et les organisations** afin de recueillir leur retex et leur vision.



Enseignements clés de l'étude

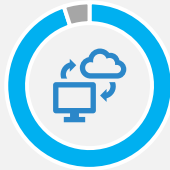
Bien que les fournisseurs européens couvrent la majorité des catégories de solutions DevSecOps, les fournisseurs américains détiennent les parts de marché les plus importantes à l'échelle mondiale et continuent, de ce fait, d'influencer les tendances globales du marché

Des fournisseurs qui structurent le marché par la plateformisation



77 %
des 13 fournisseurs
interrogés
proposent une
solution sous forme
de plateforme

Une forte tendance à la
consolidation des services



95 %
sont SaaS-only
ou hybride, sur
les 116 solutions
identifiées

Une large adoption du Cloud
par les fournisseurs



77 %
des fournisseurs
interrogés utilisent
à la fois des
composants
propriétaires et des
composants open
source existants

L'open source est utilisé pour
compléter les offres commerciales



100 %
des fournisseurs interrogés
intègrent l'IA dans leurs
solutions

L'IA s'impose comme une
tendance inévitable

Des organisations encore confrontées à des problématiques historiques



Fatigue liée aux alertes

Taux élevé de faux positifs, notamment
pour les outils de SAST*
Complexité de la remédiation, notamment
pour les scanners de secrets et les résultats
de scans SCA**
Prolifération des outils de sécurité



Un ensemble fourni et
divers de réglementations,
à opérationnaliser

Les cadres réglementaires actuels restent
formulés à haut niveau, fournissant des
orientations opérationnelles limitées quant
à la mise en œuvre des exigences.



IA : niveau d'attente
élevée vs. vraie maturité

Les capacités actuelles de l'IA dans le
DevSecOps restent encore limitées, alors
que les attentes en matière de gains
d'efficacité, notamment via la remédiation
automatisée, sont très fortes.

En conséquence, de nombreuses
organisations rencontrent des difficultés à
traduire ces exigences en pratiques
concrètes, en particulier lorsqu'elles
manquent d'expertise sur le sujet.



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

MESSAGES CLÉS DE L'ANSSI POUR L'ÉCOSYSTÈME NUMÉRIQUE



Messages clés de l'ANSSI pour l'écosystème numérique

Les orientations suivantes ont pour objectif de renforcer la maturité, la pertinence et la compétitivité des solutions DevSecOps du marché de l'Union Européenne (chaque orientation est détaillée dans les pages suivantes)

01

Développer des synergies et des partenariats entre les offres DevSecOps de l'UE en s'appuyant sur des appels à projets

02

Intégrer les exigences réglementaires de sécurité dès la conception d'un produit

03

Développer une offre DevSecOps adaptée aux TPE/PME/ETI

04

Développer une offre européenne de sécurisation de l'IA (MLSecOps*)

05

S'appuyer sur les communautés existantes pour étendre l'adoption du DevSecOps

* *Secure machine learning operations (se reporter à ce document [OpenSSF_MLSecOps_Whitepaper.pdf](#))*



Messages clés de l'ANSSI pour l'écosystème numérique

#1 - Développer des synergies et des partenariats entre les offres DevSecOps de l'UE en s'appuyant sur des appels à projets

Message clé

- ❑ L'ANSSI invite les offreurs de solutions DevSecOps à **structurer une offre européenne compétitive**, en s'appuyant notamment sur les dispositifs de financement existants. Ces cadres permettent de soutenir le développement de **plateformes DevSecOps modulaires** reposant sur des **standards ouverts** (formats et protocoles), favorisant l'intégration de composants issus de différents acteurs européens.
- ❑ **Le développement et la montée en maturité de capacités en matière d'intelligence artificielle (IA) constitue également un levier structurant.** Des partenariats entre éditeurs DevSecOps, acteurs de la recherche et fournisseurs de technologies d'IA sont encouragés afin de développer des capacités avancées de détection de vulnérabilités, d'automatisation de la remédiation et d'aide à la génération de code sécurisé.

Leviers et ressources disponibles

- ❑ La constitution de consortiums contribue au développement de synergies entre acteurs. Les éditeurs logiciels sont encouragés à **constituer de tels consortiums** afin de répondre aux **appels à projets européens** issus de programmes tels que *Horizon Europe* [EU|FR] et *Digital Europe* [EU|FR]. Ils peuvent **contacter l'ANSSI** dans le cadre du dispositif centre national de coordination (NCC-FR) :

ncc-fr.anssi@ssi.gouv.fr

- Exemple d'appel à projet :

[Approaches and tools for security in software and hardware development and assessment | Horizon-europe.gouv.fr](https://horizon-europe.gouv.fr/)



Messages clés de l'ANSSI pour l'écosystème numérique

#2 - Intégrer les exigences réglementaires de sécurité dès la conception d'un produit

Message clé

- ❑ L'ANSSI recommande aux développeurs de logiciels d'**intégrer les exigences réglementaires** dès les **premières phases du cycle de développement**, conformément aux principes de *security by design* et de *security by default*.
- ❑ Cela implique notamment :
 - la mise en œuvre de **processus continus de gestion des vulnérabilités**, incluant l'identification, la correction et le suivi des failles de sécurité ;
 - la production et la maintenance de **nomenclatures logicielles (SBOM)** assurant la traçabilité des composants ;
 - la **sécurisation par défaut des configurations logicielles et des environnements d'exécution** ;
 - la mise en place de **mécanismes de journalisation et d'audit** permettant de répondre aux obligations de notification d'incidents ;
 - la définition explicite des **durées de support et des engagements de maintien en condition de sécurité**.
- ❑ Ces mesures, priorisées en fonction de la maturité organisationnelle et de l'exposition, doivent être intégrées dans les chaînes CI/CD* à travers des **mécanismes automatisés** tels que l'analyse de composition logicielle (SCA**), les tests de sécurité (SAST/DAST***), la gestion sécurisée des secrets et l'implémentation de politiques de conformité.

Leviers et ressources disponibles

- ❑ Les développeurs sont invités à s'appuyer sur les **publications de référence** afin de traduire les exigences réglementaires - en particulier celles du *Cyber Resilience Act* (CRA) - en mesures techniques concrètes.
 - **ANSSI** : [Les essentiels - DevSecOps](#)
 - **ENISA** : [SBOM landscape analysis – Towards an implementation guide](#)
 - **Commission européenne** : [Guide de mise en pratique du CRA](#)
- ❑ Les développeurs sont invités à **suivre** et à **s'impliquer** dans les **travaux de normalisation** pour la mise en œuvre du *Cyber Resilience Act* (CRA).
 - L'ANSSI sera présente au prochain événement [CRA Standards Unlocked - EU Tour in Paris | Cyberstand](#).



Messages clés de l'ANSSI pour l'écosystème numérique

#3 - Développer une offre DevSecOps adaptée aux TPE/PME/ETI*

Message clé

- ❑ Demain, la directive européenne **NIS 2 régulera des écosystèmes entiers**. Elle imposera notamment aux EE (entités essentielles) et EI (entités importantes) de sécuriser leur chaîne d'approvisionnement, notamment par la voie contractuelle qui les lie à leurs fournisseurs et prestataires. Beaucoup d'**entreprises de services du numérique (ESN)** offrant ou assurant la maintenance de logiciels seront donc soit **directement régulées**, soit **indirectement contraintes par effet de chaîne**.
- ❑ L'ANSSI encourage les offreurs de solutions DevSecOps à adopter des modèles de diffusion **adaptés aux TPE, PME et ETI*** (p. ex. selon une stratégie commerciale *freemium*).

Leviers et ressources disponibles

- ❑ Les acteurs sont encouragés à mobiliser les opportunités de financement offertes afin de répondre aux **appels à projets européens** issus de programmes tels que *Horizon Europe* [EU|FR], *Digital Europe* [EU|FR]. Ils peuvent **contacter l'ANSSI** dans le cadre du dispositif centre national de coordination (NCC-FR) :

ncc-fr.anssi@ssi.gouv.fr

- Exemple d'appel à projet :

[ECCC to finance EUR 390 million in cybersecurity projects under Digital Europe Programme for 2025-2027](#)

- ❑ Plus particulièrement destiné aux start-up du numérique, le guide de l'ANSSI « [Guide de cybersécurité à l'usage des start-up du numérique](#) » pourra aussi être d'intérêt pour des TPE, PME et ETI*.



#4 - Développer une offre européenne de sécurisation de l'IA (MLSecOps)

Message clé

- ❑ L'essor des systèmes d'intelligence artificielle impose l'émergence d'une **approche MLSecOps** intégrant nativement les enjeux de sécurité et de conformité réglementaire, **non seulement au niveau des modèles**, mais également dans **leurs usages opérationnels, notamment via des agents IA intégrés aux chaînes CI/CD**.
- ❑ L'ANSSI encourage les offreurs à développer des solutions permettant de traiter les risques pesant sur :
 - **les systèmes d'IA** : l'empoisonnement des données, les attaques adverses, l'extraction de modèles ou encore les fuites d'informations sensibles ;
 - **les usages d'agent IA dans la production logicielle** : la génération de code vulnérable ou non maîtrisé (hallucination de dépendances, absence de contrôle qualité), la fuite de secrets ou d'informations sensibles via des prompts générés, la dilution des responsabilités dans les processus de développement, etc.
- ❑ Par ailleurs, les solutions développées devront **faciliter la mise en conformité avec le règlement européen sur l'intelligence artificielle (AI Act)**, notamment grâce à des capacités automatisées d'auditabilité, de traçabilité et de documentation des modèles.

Leviers et ressources disponibles

- ❑ Les acteurs sont encouragés à mobiliser les opportunités de financement offertes afin de répondre aux **appels à projets européens** issus de programmes tels que *Horizon Europe* [EU|FR], *Digital Europe* [EU|FR]. Ils peuvent **contacter l'ANSSI** dans le cadre du dispositif centre national de coordination (NCC-FR) :

ncc-fr.anssi@ssi.gouv.fr

- Exemple d'appel à projet :

[Enhancing the Security, Privacy and Robustness of AI Models and Systems \(SecureAI\) | Horizon-europe.gouv.fr](https://horizon-europe.gouv.fr/enhancing-the-security-privacy-and-robustness-of-ai-models-and-systems-secureai)



#5 - S'appuyer sur les communautés existantes pour étendre l'adoption du DevSecOps

Message clé

- ❑ L'ANSSI invite les acteurs publics et privés, **utilisateurs de solutions DevSecOps**, à s'appuyer sur les **communautés existantes** afin de favoriser le partage de bonnes pratiques et l'élévation du niveau global de maturité en matière de DevSecOps.
- ❑ La structuration d'une **communauté nationale dédiée au DevSecOps**, associant **acteurs publics, industriels et académiques**, pourrait constituer un levier supplémentaire pour favoriser l'émergence de standards communs et renforcer la cohérence des initiatives.

Leviers et ressources disponibles

- ❑ Les organisations sont **encouragées à participer activement aux travaux menés par des relais de connaissance et d'influence** :
 - **Relais techniques** qui fournissent à l'écosystème des plateformes et outils d'automatisation et de gestion de configuration pour déployer, orchestrer et maintenir des infrastructures informatiques de manière reproductible et sécurisée via du code* (p. ex. [Ansible Galaxy](#), [Puppet Forge](#), [pyinfra](#)) ;
 - **Relais de gouvernance** qui contribuent à la diffusion de référentiels et de retours d'expérience en matière de sécurité applicative et de DevSecOps (p. ex. [CESIN](#)*, le [CLUSIF](#)** , le [Campus Cyber](#)).



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

ÉTUDE DE MARCHÉ S-SDLC/DEVSECOPS



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

INTRODUCTION



Contexte et objectifs de l'étude

Contexte

Le paysage du **S-SDLC** et du **DevSecOps** est structuré par des pressions significatives, résultant à la fois de **menaces systémiques récurrentes ciblant la chaîne d'approvisionnement** (p. ex. attaques SolarWinds, XZ Utils) et du **renforcement des exigences réglementaires** (p. ex. Cyber Resilience Act - CRA).

Dans le cadre de la **mission de politique industrielle de l'ANSSI**, la division **Industrie et Technologie** doit renforcer ses **connaissances du marché S-SDLC et DevSecOps** afin d'identifier d'éventuelles lacunes nécessitant une intervention au titre de la politique industrielle.

Objectifs

Les principaux objectifs de cette étude sont de :

- Fournir une vision globale du **marché S-SDLC et DevSecOps européen**
- Suggérer des recommandations concrètes **pour répondre aux lacunes identifiées sur le marché**

Cette étude vise également à fournir des lignes directrices pour **sécuriser les pipelines CI/CD**.



Concepts clés et définitions

Définitions de S-SDLC, DevSecOps et MLSecOps

Le **Secure Software Development Life Cycle (S-SDLC)** fournit un cadre structuré intégrant les activités de sécurité à chaque phase du développement logiciel.

DevSecOps est une approche culturelle et technique qui intègre les pratiques de sécurité au sein des flux de travail DevOps, un paradigme visant l'unification et l'automatisation du développement logiciel (Dev) et de l'administration des infrastructures informatiques (Ops), notamment en ce qui concerne l'administration système.

MLSecOps applique ces mêmes principes pour le développement à large échelle et la gestion de systèmes d'intelligence artificielle (IA) et de *machine learning* (ML) en intégrant la sécurité, la conformité et la gestion des risques tout au long du cycle de vie des modèles (données, entraînement, déploiement et supervision).



Concepts clés et définitions

12 catégories de solutions sont incluses dans le périmètre de l'étude et répondent à 4 besoins clés en matière de sécurité

I. Étendre et améliorer les capacités de détection au sein du SDLC

1. **Static Application Security Testing (SAST)** – Identifie les vulnérabilités en analysant le code source ou les binaires, sans exécuter l'application.
2. **Dynamic Application Security Testing (DAST)** – Détecte les problèmes de sécurité en testant l'application en fonctionnement, en simulant des attaques externes.
3. **Interactive Application Security Testing (IAST)** – Combine les techniques SAST et DAST en instrumentant l'application pour analyser son comportement en temps réel.
4. **Runtime Application Self-Protection (RASP)** – Surveille et protège l'application en détectant et en bloquant les menaces lors de l'exécution.
5. **Software Composition Analysis (SCA)** – Détecte les vulnérabilités et les risques liés aux composants logiciels tiers et open source, en s'appuyant généralement sur un **Software Bill of Materials (SBOM)** pour fournir un inventaire structuré de tous les composants, dépendances et risques associés.
6. **Scanner IaC (Infrastructure As Code)** – Analyse les modèles IaC (p. ex. Terraform, CloudFormation) afin de détecter les mauvaises configurations et les risques de sécurité avant le déploiement.
7. **Artefact scanner** – Examine les artefacts compilés (p. ex. conteneurs, binaires) afin de détecter les vulnérabilités connues et les problèmes de conformité.

II. Assurer une gestion sécurisée des secrets dès la phase de développement

8. **Solutions de détection et gestion des secrets** – Stockent de manière sécurisée les données sensibles et détectent les secrets codés en dur ou exposés dans le code et les pipelines.

III. Standardiser et automatiser les processus de sécurité

9. **Solutions de modélisation des menaces** – Aident à identifier et évaluer les menaces potentielles durant la phase de conception du cycle de vie logiciel.
10. **Solutions de signature d'artefacts** – Garantissent l'authenticité et l'intégrité des artefacts logiciels grâce à des signatures cryptographiques.
11. **Application Security Posture Management (ASPM)** – Agrège et priorise les résultats de sécurité issus de différentes solutions afin de fournir une visibilité sur la posture de sécurité des applications.

IV. Renforcer les compétences et la maturité en matière de S-SDLC

12. **AppSec/DevSecOps Training Platforms** – Fournit des formations axées sur les pratiques de développement sécurisées et l'intégration de la sécurité dans les flux de travail DevOps.



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

ANALYSE DU MARCHÉ ET TENDANCES STRUCTURELLES



Principaux facteurs d'accélération de l'adoption du S-SDLC et du DevSecOps

L'intégration de la sécurité progresse en raison de l'augmentation des risques liés à la *supply chain* logicielle, de l'évolution des cadres réglementaires et de l'essor des modèles de développement cloud-native.

1 – Augmentation des risques



- Exposition croissante aux risques liés à la *supply chain* logicielle.
- Incidents majeurs impliquant des dépendances open source, des pipelines CI/CD et de l'Infrastructure-as-Code.

→ Les contrôles de sécurité appliqués trop tardivement ne sont plus suffisants, ce qui conduit à intégrer la sécurité plus en amont dans le SDLC.

2 – Évolution réglementaire



- Les exigences en matière de réglementation et de conformité se durcissent.
- Des cadres comme le RGPD, NIS2, DORA et le Cyber Resilience Act introduisent des exigences renforcées en matière de sécurité dès la conception, de gestion des risques de la *supply chain* et de traçabilité sur l'ensemble du SDLC.

→ Le DevSecOps évolue, passant du statut de bonne pratique à celui d'obligation réglementaire.

3 – Adoption cloud



- Adoption de modèles de livraison cloud-native et automatisés.
- la virtualisation, les conteneurs et leurs orchestrations (p. ex. Kubernetes, Docker Swarm), les microservices et l'*Infrastructure-as-Code* (IaC) ont transformé le développement logiciel.

→ Le DevSecOps s'impose ainsi comme une extension naturelle des modèles opérationnels Cloud, et non plus comme une pratique marginale.

En conséquence, le marché du DevSecOps et du S-SDLC connaît une croissance soutenue*, portée par l'expansion des cas d'usage, la consolidation des plateformes et des investissements induits par les évolutions réglementaires.

En savoir plus sur la
méthodologie
d'analyse de marché



*Sources : Peak Evolv Edge Hub (TCAC^[18] : 12,98 %), Grand View Research (TCAC : 13,2 %), Future Market Insights (TCAC : 16,9 %), Mordor Intelligence (TCAC : 23,6 %).
[18] Taux de croissance annuel composé



Un besoin de répondre aux risques et menaces en évolution

L'adoption du DevSecOps s'accélère en réponse à l'évolution des risques cyber, combinant des enjeux persistants de gestion des vulnérabilités, la multiplication des attaques sur la *supply chain* logicielle et l'émergence de nouveaux risques liés aux évolutions technologiques.

Gestion des vulnérabilités :
un risque historique et
persistant

- **Volume croissant de vulnérabilités connues**
- **Vulnérabilités connues** restant **actives en production** pendant des périodes prolongées
- **Les composants legacy et les dépendances obsolètes** accumulent des risques techniques à long terme

La *supply chain* logicielle, aujourd'hui largement fondée sur des composants **open source**, est **ciblée** par des groupes malveillants et des acteurs étatiques, notamment pour implanter des backdoors :

- **2024 - XZ Utils** : Des millions d'appareils Linux exposés, en raison d'une backdoor insérée dans la *supply chain* de la bibliothèque Linux XZ Utils lors d'une attaque coordonnée.
- **2025 - Shai-Hulud 2.0** : Ver auto-replicant de *supply chain* npm ayant compromis plus de 700 paquets npm, y compris des bibliothèques populaires telles que Zapier et PostHog.
- **2025 - React2Shell** : Vulnérabilité permettant une exécution de code à distance non authentifiée, affectant presque tous les sites web créés avec des versions récentes de React.js et de Next.js.

En savoir plus sur les scénarios d'attaque de la chaîne CI/CD

Risques émergents liés à
l'évolution technologique

- **Cloud** : Mauvaises configurations, *workloads* dynamiques et éphémères.
- **Intelligence artificielle** : vulnérabilités dans les pipelines et modèles IA/ML, code généré par IA non sécurisé, empoisonnement de données et manipulation des modèles.
- **Cryptographie post-quantique** : risques de déchiffrement futur des données actuellement chiffrées.



Aperçu des réglementations cybersécurité obligatoires pour le S-SDLC

Réglementation/Cadre

Qui est affecté ?

(Types d'organisations, secteurs, pays)

Type de texte

Europe



DORA 2022*

Entités financières (banques, assurances, sociétés d'investissement, éditeurs de paiements, fintech) opérant dans l'UE ainsi que des fournisseurs de services TIC** critiques les soutenant.

Règlement et directive européens



CRA***

Secteur des produits numériques et logiciel (fabricants, éditeurs de logiciels, distributeurs) plaçant des produits sur le marché européen.

Règlement européen



LPM****

Opérateurs d'importance vitale (OIV) en France (énergie, transports, télécoms, défense, santé), pour renforcer leur résilience, via notamment la sécurisation de leurs systèmes d'information et chaîne d'approvisionnement.

Loi nationale



NIS2 2022

Entités essentielles et importantes dans des secteurs critiques (énergie, transports, santé, finance, administration publique, etc.) opérant dans l'UE.

Directive européenne



PLD*** 2024/25**

Secteur des produits logiciels (fabricants, éditeurs de logiciels, distributeurs) plaçant des produits sur le marché européen.

Directive européenne



AI Act

Secteur de l'intelligence artificielle (fournisseurs, intégrateurs et distributeurs de systèmes) plaçant ses produits sur le marché européen.

Règlement européen

Hors Europe



U.S. Executive Order 14028

Supply chain logicielle du gouvernement fédéral (éditeurs de logiciels, fournisseurs de services TIC*) délivrant à des agences fédérales américaines.

Décret exécutif américain



SLACIP Act

Infrastructures critiques (opérateurs dans 11 secteurs, dont l'énergie, les communications, la santé) opérant en Australie.

Loi australienne



PCI-DSS

Secteur du traitement des paiements (commerçants, prestataires de services traitant les données des titulaires de carte) opérant à l'échelle mondiale.

Norme de sécurité



**Cyber Security Agency
Advisory Singapore**

Secteur des logiciels et des services numériques (éditeurs de logiciels, fournisseurs cloud/SaaS, fournisseurs TIC*) opérant à Singapour.

Recommandations

**** Loi de programmation militaire

***** Product Liability Directive



Exigences réglementaires favorisant le développement logiciel sécurisé

Les réglementations redéfinissent les priorités du S-SDLC et du DevSecOps

Synthèse des réglementations 

À travers l'Europe, les États-Unis et au-delà, les gouvernements convergent vers une position commune : **la sécurité doit être intégrée dans le cycle de vie logiciel dès la conception.**

Thèmes	Exigences principales	Sources
Développement sécurisé & sécurité applicative	<ul style="list-style-type: none"> Concevoir des systèmes selon les principes « secure-by-design » et « secure-by-default », en protégeant les pipelines CI/CD contre toute altération et en appliquant des pratiques de codage sécurisé pour prévenir les vulnérabilités courantes. Contrôler et surveiller les environnements de développement, en mettant en place des protections robustes pour empêcher toute modification non autorisée. Assurer une traçabilité complète des modifications, conserver des logs d'audit exhaustifs et démontrer que le logiciel est conçu, testé et publié selon des processus rigoureusement encadrés et sécurisés. 	EO 14028, NIS2, CRA, LPM, PCI-DSS, Loi sur l'AI, SLACIP
SCA/SBOM & Transparence logicielle	<ul style="list-style-type: none"> Maintenir une transparence totale sur les composants qui composent les logiciels, générer et mettre à jour les SBOMs, documenter toutes les dépendances tierces et open source, et vérifier en permanence la provenance des composants et leur intégrité. Suivre les vulnérabilités affectant ces composants, tracer tous les changements dans la documentation technique et fournir des SBOMs aux régulateurs ou clients sur demande. Maintenir un historique clair des mises à jour pour démontrer une gestion responsable de la composition logicielle. 	EO 14028, NIS2, CRA, Conseil CSA, DORA, PLD
Gestion des secrets et protection des actifs sensibles	<ul style="list-style-type: none"> Mettre en place des contrôles d'accès stricts, chiffrer les actifs sensibles et empêcher l'exposition de secrets dans les dépôts de code ou les pipelines CI/CD. Assurer un stockage sécurisé et une rotation des éléments d'authentification, éliminer les identifiants codés en dur et déployer des mécanismes de surveillance capables de détecter toute mauvaise utilisation ou accès non autorisé. Démontrer que les actifs sensibles sont constamment protégés dans des conditions contrôlées et auditables. 	PCI-DSS, NIS2, EO 14028, LPM



Influence limitée de la recherche sur le S-SDLC/DevSecOps

Les publications portant sur le S-SDLC et le DevSecOps révèlent un paysage de recherche limité et fragmenté.

Des publications avec des analyses fragmentées plutôt que des *frameworks* complets

L'analyse des **publications de recherche révèle une tendance** : la plupart des papiers de recherches analysés se concentrent sur des **aspects techniques isolés** tels que l'enrichissement des SBOM, l'analyse de la *supply chain* logiciel, l'automatisation CI/CD ou le MLSecOps, tandis que les modèles holistiques restent **rarement explorés**. Malgré la pression réglementaire croissante, le domaine repose encore sur des approches descriptives avec **peu de tests en conditions réelles**, ce qui conduit à des **éclairages fragmentés** plutôt qu'à un cadre S-SDLC unifié.

Aperçu de la
Recherche
académique



Thèmes principaux de la recherche

Sur la base de notre revue des publications de recherche et de l'industrie, quatre thèmes principaux ont été identifiés :

1. **Sécurité dans le SDLC / DevSecOps**
2. **Pipeline ML/IA (MLSecOps)**
3. **Sécurité de la supply chain logicielle**
4. **Sécurité logicielle en pratique : Connaissances et motivation**

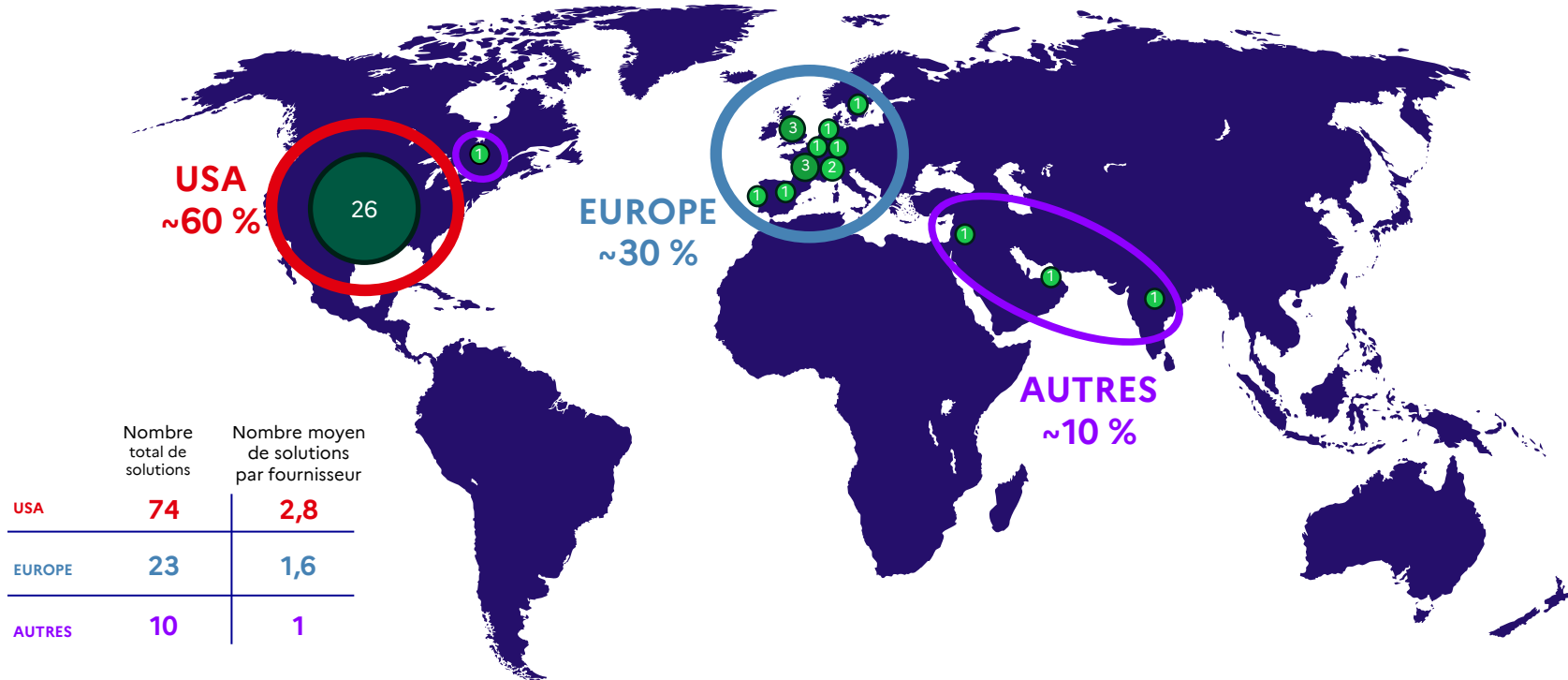


Cette sélection d'articles, incluant des publications non évaluées par les pairs, demeure sujette à caution et ne reflète que partiellement l'état de l'art en DevSecOps, une analyse plus exhaustive n'ayant pu être menée en raison des contraintes temporelles de l'étude.



Distribution des fournisseurs S-SDLC / DevSecOps par pays

La répartition géographique des fournisseurs DevSecOps identifiés met en évidence la prédominance des fournisseurs et solutions basés aux États-Unis





Tendances impulsées par les États-Unis et une évolution vers des plateformes de sécurité intégrées

Le marché du S-SDLC / DevSecOps est largement dominé par des fournisseurs américains et évolue vers des solutions plateformes, principalement SaaS, enrichies par des composants open source et des capacités d'intelligence artificielle émergentes.

Observations générales du marché

1

Les plateformes de sécurité* vont façonner le marché

Le marché évolue vers des plateformes de sécurité complètes qui regroupent **plusieurs solutions DevSecOps en un seul écosystème**. Cela simplifie non seulement l'adoption et la gestion, mais permet aussi aux fournisseurs d'**élargir progressivement leurs capacités**, créant ainsi des offres plus complètes pour les clients.

2

Les solutions SaaS deviennent la norme

Le déploiement SaaS et hybride deviennent rapidement **le modèle par défaut** pour les solutions DevSecOps. Ils offrent un délai de mise en œuvre plus rapide, réduisent les coûts de maintenance et permettent des mises à jour transparentes, permettant aux organisations d'adopter des solutions de sécurité sans investissements importants dans les infrastructures. Cette transition **soutient l'essor de plateformes de sécurité entièrement basées sur le Cloud**.

3

Les fournisseurs comptent fortement sur l'open source

Les composants open source restent au cœur des solutions DevSecOps, permettant aux fournisseurs d'implémenter rapidement de nouvelles fonctionnalités tout en conservant leur flexibilité. Combinée à **des acquisitions opportunistes**, cette stratégie aide les plateformes à **combler efficacement leurs lacunes fonctionnelles, renforçant leur proposition de valeur et accélérant la consolidation du marché**.

4

L'IA sera intégrée à chaque étape du S-SDLC / DevSecOps

L'IA est de plus en plus intégrée dans les solutions DevSecOps, suite à l'augmentation de l'utilisation mondiale de l'IA et **aux attentes élevées** des organisations. Les fournisseurs promeuvent des capacités avancées telles que la **réduction des faux positifs, l'assistance à la remédiation et l'automatisation agentique**. Cependant, ces promesses varient **considérablement en maturité**, ce qui rend essentiel de **différencier les capacités réelles des affirmations marketing**.



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

FOCUS SUR LA DYNAMIQUE DE LA DEMANDE



Défis de maturité DevSecOps identifiés par les organisations interrogées

Malgré une adoption croissante du DevSecOps, les faux positifs et la complexité des processus de remédiation freinent la montée en maturité des organisations.



Quels sont les principaux enjeux à traiter en priorité et les solutions privilégiées ?

Les faux positifs et la complexité de remédiation sont les principaux irritants – Les solutions SAST génèrent trop de faux positifs. La **complexité des remédiations*** est particulièrement remontée pour les résultats d'analyse des scanners de secrets (applications legacy) et pour les résultats d'analyse SCA (impliquant parfois une restructuration importante du code).

La plateformes est privilégiée – Le « *best of breed* » reste envisagé lors du choix de solutions, en attendant que des plateformes matures proposent des solutions efficaces sur tous les segments.

Les premières solutions adoptées sont les SCA, SAST et les scanners de secrets – DAST et IAST arrivent ensuite, après l'adoption des premières solutions.



Les fonctionnalités d'intelligence artificielle sont-elles matures ?



Les fonctionnalités sont en cours de maturation – Les tests menés par les organisations montrent que les fonctionnalités d'IA intégrées dans les solutions DevSecOps ne sont pas encore pleinement matures. Néanmoins, de fortes attentes subsistent quant à leur évolution, afin de rendre les opérations des organisations plus efficaces (p. ex., pour le tri et la correction des vulnérabilités).

Les LLM utilisés par les fournisseurs sont génériques – Les organisations attendent des LLM^[19] spécialisés en cybersécurité pour obtenir des recommandations plus pertinentes (plutôt que des modèles génériques tels que Claude, Mistral, Chat-GPT, etc.).



Quel est l'impact des réglementations sur les organisations ?

La sécurité comme moteur principal – La plupart des organisations ont commencé leur démarche DevSecOps avant tout pour renforcer leur posture de sécurité, plutôt que pour répondre aux exigences réglementaires.

Un problème d'interprétation – Les organisations considèrent que les réglementations actuelles manquent de spécificités concernant le DevSecOps, ce qui conduit à des interprétations qui pourraient ne pas être alignées avec les attentes réglementaires.

Quelles sont les attentes envers l'éco-système ?



Fournir un cadre de référence DevSecOps – Soutenir la mise en œuvre des exigences réglementaires en fournissant des directives opérationnelles, des recommandations prioritaires et des outils d'auto-évaluation alignés sur les bonnes pratiques DevSecOps.

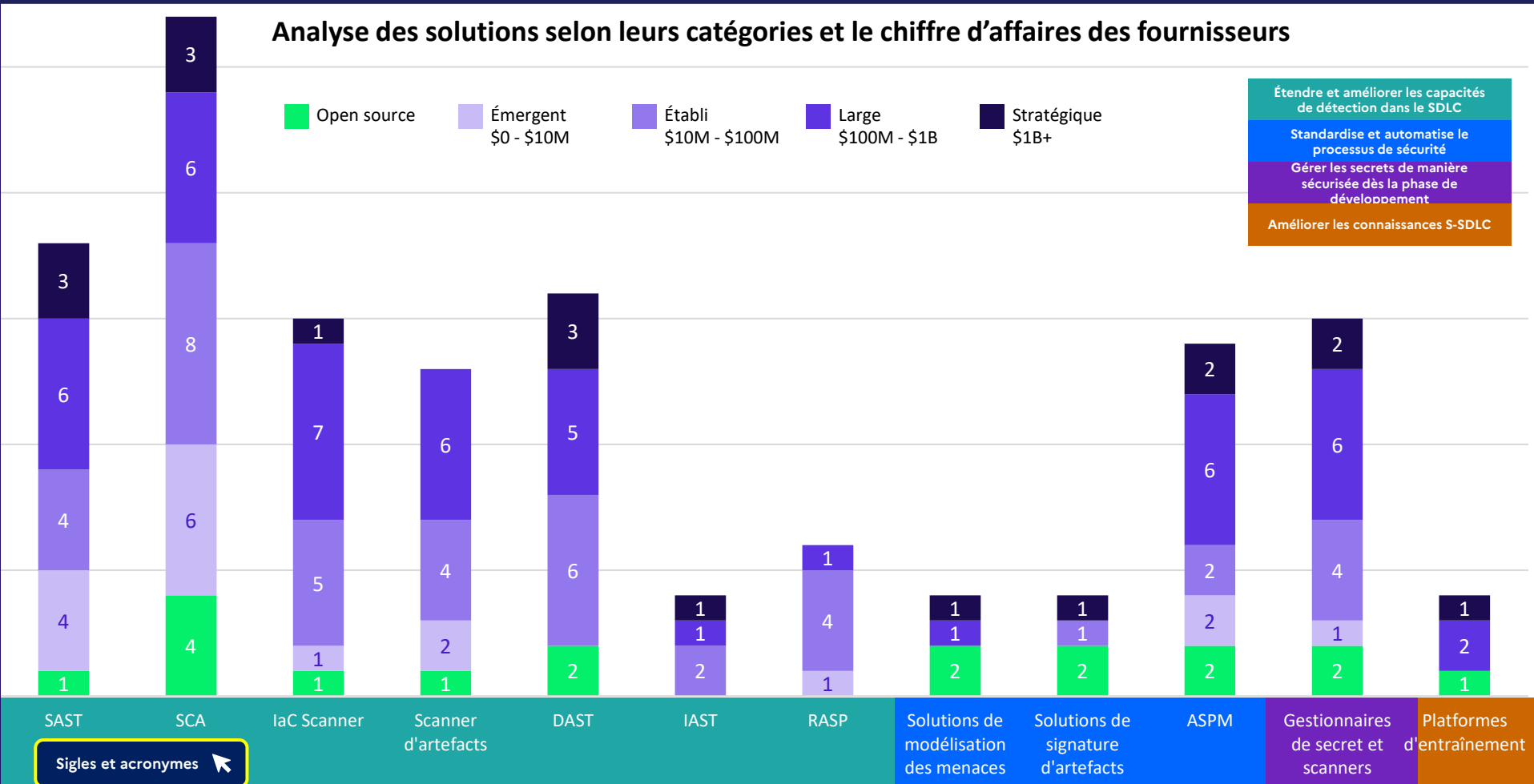
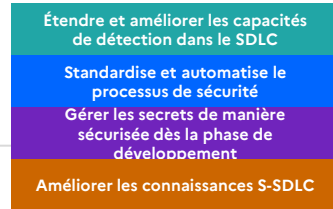
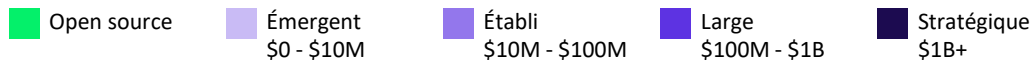
Animer / soutenir une communauté DevSecOps – Favoriser le partage de connaissances, de retours d'expériences et de bonnes pratiques afin d'élever le niveau de maturité de l'écosystème.



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?
FOCUS SUR LA DYNAMIQUE DE L'OFFRE

L'offre de solutions S-SDLC est diversifiée, avec une large gamme de tailles de fournisseurs

Analyse des solutions selon leurs catégories et le chiffre d'affaires des fournisseurs



Sigles et acronymes



Répartition des fournisseurs

96 solutions S-SDLC / DevSecOps ont été identifiées parmi 42 fournisseurs

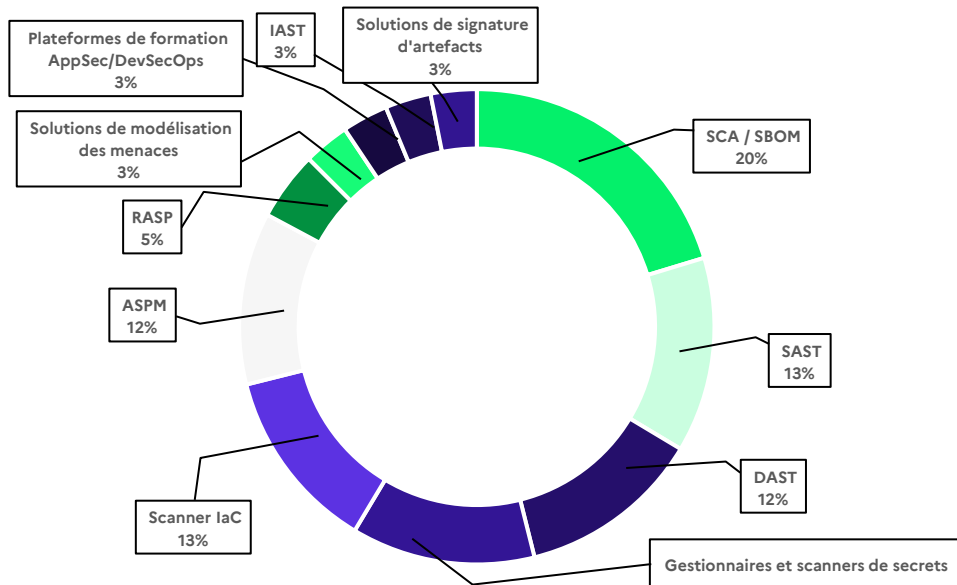


SCA/SBOM est la catégorie la plus représentée parmi l'ensemble des solutions DevSecOps



Le déploiement hybride est devenu le modèle principal adopté par les fournisseurs DevSecOps

% de solution par catégorie S-SDLC



~30 % fournissent des **solutions SaaS exclusivement cloud**, ne nécessitant aucun déploiement sur site



~65 % proposent un **hébergement hybride**, combinant plateformes cloud et infrastructures sur site.



~5 % proposent **uniquement un hébergement sur site** et pas de plateformes SaaS





Paysage des solutions : États-Unis vs Europe

Le marché américain se distingue par une offre DevSecOps mature et diversifiée, tandis que l'Europe conserve une présence significative dans la majorité des segments, participant ainsi à la structuration d'un écosystème large et en expansion continue.



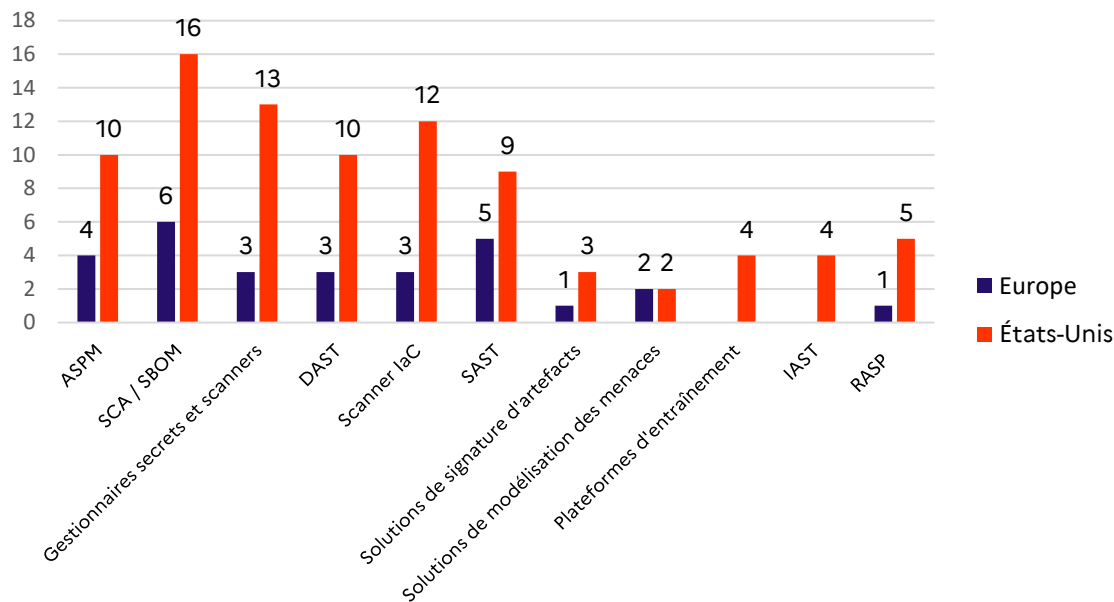
Maturité des écosystèmes

Etats-Unis : une tendance à la consolidation amorcée depuis longtemps, avec de nombreux fournisseurs proposant des solutions « **plateformisées** ».



Présence européenne

Europe : un marché fragmenté qui couvre néanmoins l'ensemble des besoins en sécurité des pipelines CI/CD, avec une forte représentation des solutions **SCA/SBOM** et **SAST**.





Focus sur les fournisseurs européens

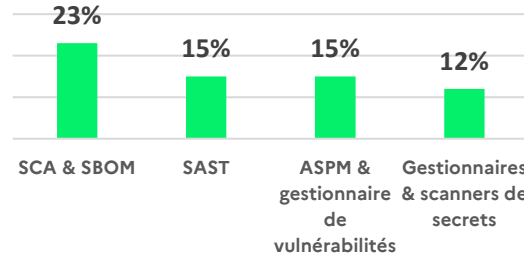


13

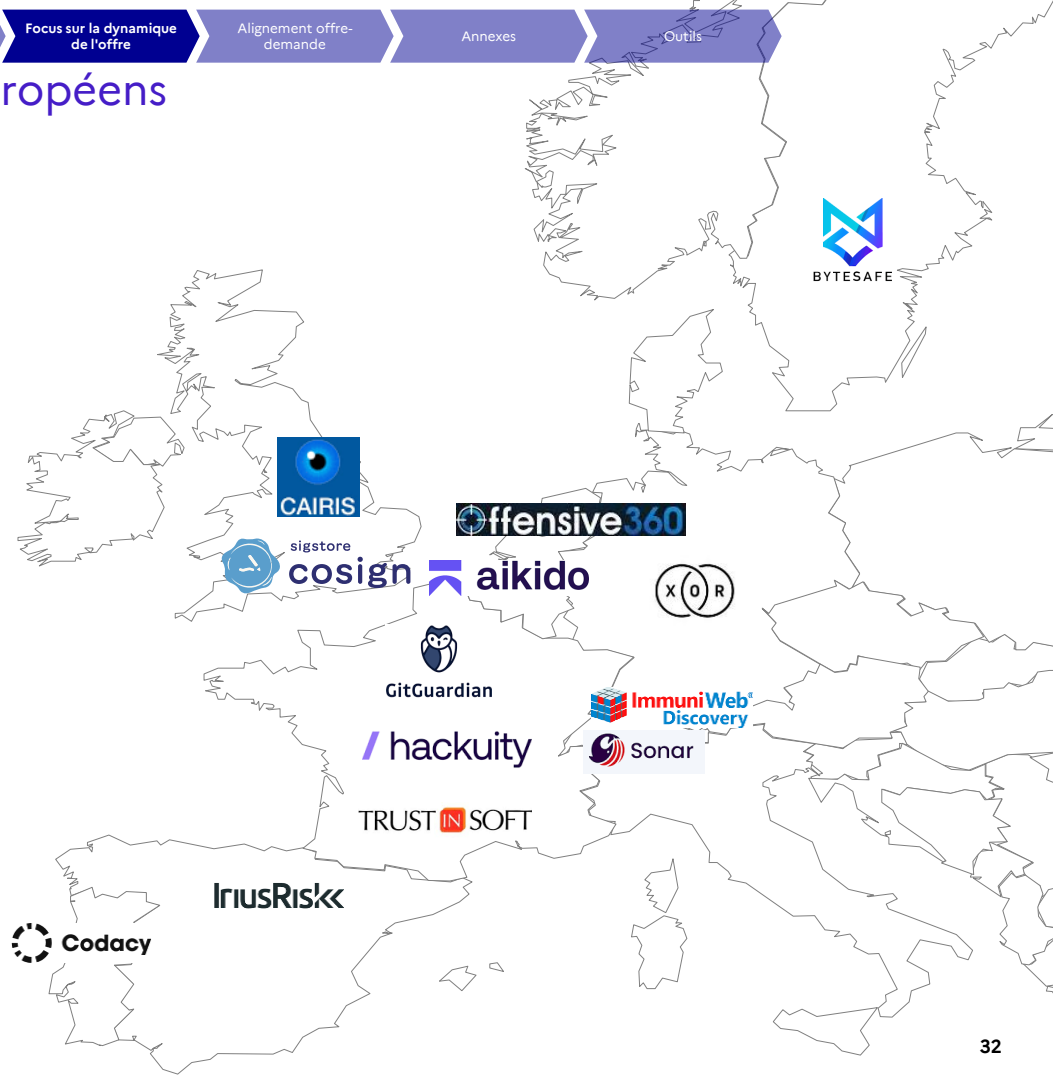
fournisseurs européens identifiés sur le marché du S-SDLC et du DevSecOps

- 1 Les solutions **SCA/SBOM** et **SAST** sont légèrement plus représentés que l'**ASPM** par rapport aux tendances mondiales
- 2 Trois fournisseurs **Français** proposent des solutions d'**ASPM (gestionnaire de vulnérabilités)**, de **Secret Managers & Scanners** et de **SAST**
- 3 **62 %** des fournisseurs européens proposent des solutions **SaaS**

Principales catégories de solutions proposées par les fournisseurs européens



Suivi par : scanner IaC et DAST 12 %, modélisation des menaces 8 %, RASP 4 %



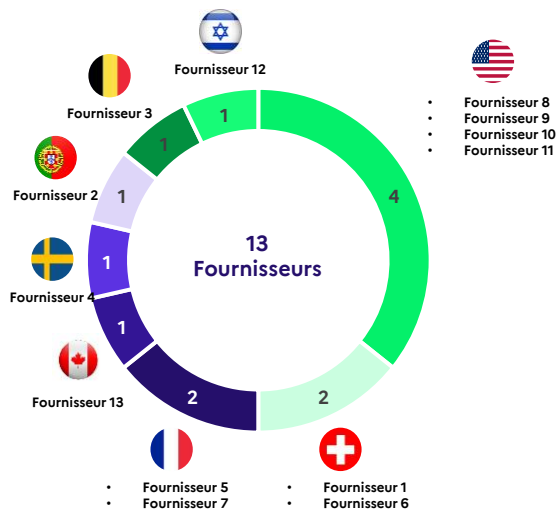


Rencontres fournisseurs

Accent sur trois catégories clés : SCA/SBOM, gestion des secrets et scanners, et ASPM

En savoir plus sur la
sélection de ces 3
catégories

Fournisseurs par pays



Catégories de solutions couvertes par région et éditeur

Parmi les 13 éditeurs interrogés, 10 proposent des solutions de SCA/SBOM (4 européennes), 7 proposent des solutions d'ASPM (4 européennes), et 3 proposent des solutions de gestionnaires et scanners de secrets (toutes européennes)

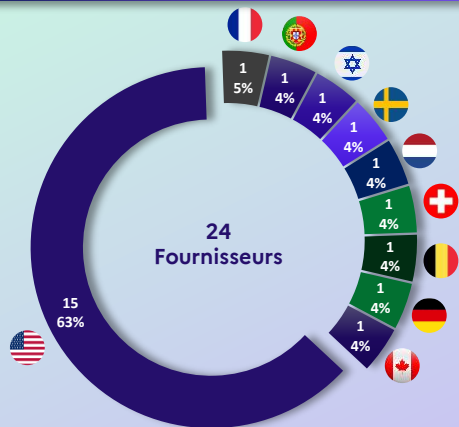
Région	SCA / SBOM	Gestionnaires et scanners de secrets	ASPM
Europe	✓	✓	
	✓		
	✓		✓
	✓		
		✓	✓
		✓	✓
USA	✓		✓
	✓		✓
	✓		
	✓		
Israël	✓		✓
Canada	✓		
Total	10	3	7



Focus – SCA / Software Bill Of Material (SBOM)

Informations clés sur les pratiques SBOM parmi les 10 fournisseurs évalués

SCA/SBOM : répartition des fournisseurs



SCA/SBOM : fournisseurs interrogés

USA	EUROPE	AUTRES
4 Fournisseurs	4 Fournisseurs	2 Fournisseurs

Observations clés



La génération de SCA/SBOM est aujourd'hui considérée comme **mature** par les fournisseurs et le principal défi porte désormais sur la gestion complète du cycle de vie des SBOM.

100 % des fournisseurs mettent l'accent sur la **gestion des SBOM**, afin de transformer ces artefacts statiques de conformité en éléments actionnables dans les processus de sécurité continus.

Convergence claire autour des **formats SBOM reconnus (CycloneDX et SPDX)**.

Intégrant la prise en compte progressive des exigences réglementaires, les feuilles de route sont de plus en plus orientées **par la transparence et la conformité**.

La gestion des SBOMs et la détection des vulnérabilités restent toutefois **partiellement intégrées** et conservent souvent des workflows, des sources de données et parcours utilisateurs distincts.



4 Cas d'usage clés pour les solutions de SCA

1. Ingestion et normalisation de SBOMS externes
2. Gestion des vulnérabilités et des licences
3. Application de politiques de sécurité internes
4. Génération d'éléments de preuve auditable



Sujets émergents

AI-BOM : Inventaire recensant tous les composants, dépendances et ressources impliqués dans la construction, l'entraînement et l'exploitation d'un système d'IA.

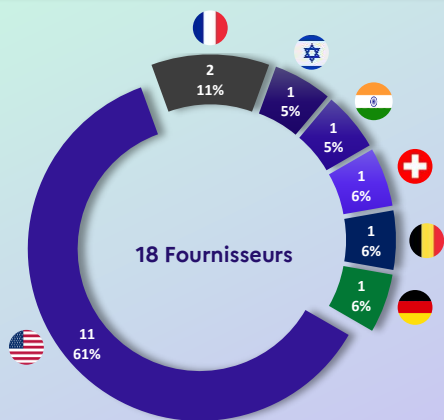
Cryptographic-BOM : Inventaire recensant tous les composants des algorithmes et bibliothèques cryptographiques utilisés dans les logiciels.



Focus – ASPM / gestion de vulnérabilités

Informations clés sur les pratiques ASPM parmi les 6 fournisseurs évalués

Répartition des fournisseurs ASPM



Fournisseurs ASPM interrogés

USA

2
Fournisseurs

EUROPE

3
Fournisseurs

Observations clés



L'ASPM s'impose progressivement comme un point central de la sécurité applicative

L'ASPM devient le **point central de la sécurité applicative**, en consolidant les résultats des solutions SAST, DAST, SCA et autres outils de sécurité.

Les fournisseurs souhaitant proposer l'ensemble des solutions DevSecOps au sein d'une plateforme unique, s'appuient sur l'ASPM comme **élément central du pilotage du niveau de risque et de sécurité**.



Principaux cas d'usages

1. Consolidation centralisée des résultats AppSec
2. Priorisation des risques entre les applications
3. Suivi de la conformité et application des politiques de sécurité
4. Visibilité sur l'ensemble du SDLC pour les développeurs et les équipes de sécurité



Sujets émergents

La remédiation automatique des vulnérabilités permet de corriger ou d'atténuer automatiquement les problèmes de sécurité détectés, afin de réduire les risques et d'optimiser le SSDLC.

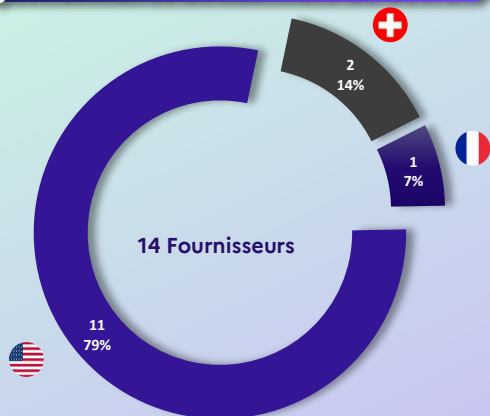
Combiner le DevSecOps et la gestion de la surface d'attaque externe : démarche visant à identifier, surveiller et réduire les vulnérabilités des actifs exposés sur Internet.



Focus – Gestionnaires & scanners de secrets

Informations clés sur les pratiques des gestionnaires et scanners de secrets parmi les 3 fournisseurs évalués

Répartition des fournisseurs de gestionnaires et scanners de secrets



Fournisseurs de gestionnaires et scanners de secrets interrogés

Solutions intégrées dans des plateformes DevSecOps

2 Fournisseurs

Gestionnaire de secrets et des identités non humaines (NHI)

1 Fournisseur

Observations clés



Les solutions de détections de secrets sont considérées comme matures par les fournisseurs, qui se concentrent désormais sur la remédiation

La détection de secrets est désormais une **capacité mature** en matière de sécurité applicative.

La plupart des fournisseurs proposent **des fonctionnalités complètes et fiables** de découverte automatisée et de surveillance continue tout au long du SDLC.

Ces solutions protègent les informations sensibles telles que les clés API, les tokens et les identifiants.



Principaux cas d'utilisation

- Détection de secrets exposés dans des dépôts publics (p. ex. GitHub).
- Détection de secret stocké dans des systèmes internes (p. ex. pipelines CI/CD, Infrastructure As Code).



Sujets émergents

L'expansion de l'automatisation et l'émergence de l'IA agentic entraîne une forte croissance des identités non humaines dans les environnements numériques. Ces entités dépendent de **secrets pour fonctionner, créant une surface d'attaque en forte expansion** qui nécessite une gouvernance dédiée.

La **gestion des secrets des identités non humaines (NHI)** désigne l'ensemble des pratiques visant à **découvrir, protéger et gérer** les secrets (clés API, tokens) utilisés par des identités non humaines (applications, services, scripts, etc.) afin de **prévenir les accès non autorisés** et de réduire les risques de sécurité dans des environnements fortement automatisés.



Observations globales des fournisseurs

Les fournisseurs interrogés s'alignent sur les tendances globales du marché



Les solutions « SaaS-first » basées sur le cloud domineront

95 % des solutions identifiées sont soit **exclusivement SaaS**, soit **hybrides** (30 % SaaS / 65 % Hybride / 7 % On-Premise).

L'On-Prem subsiste principalement pour des besoins niches ou pour répondre à des exigences réglementaires



Les plateformes de sécurité vont façonner le marché

80 % des fournisseurs interrogés visent à proposer une approche basée sur une plateforme couvrant plusieurs besoins du DevSecOps

Des solutions spécialisées ont également été identifiées dans la gestion des SBOMs, la gestion et les scans de secrets



Le marché dépend fortement de l'open source

77 % des fournisseurs DevSecOps interrogés utilisent à la fois des composants propriétaires et open source dans leurs solutions.



L'IA sera intégrée à chaque étape du S-SDLC / DevSecOps

100 % des fournisseurs interrogés intègrent l'IA dans leurs solutions, avec des cas d'usage courants tels que : la remédiation automatique, l'enrichissement des bases de données de vulnérabilités, la priorisation des vulnérabilités, la gestion des faux positifs ou la revue de code assistée par LLM.



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?
ALIGNEMENT OFFRE-DEMANDE



Alignement offre-demande

L'offre des fournisseurs est globalement alignée avec les besoins des organisations, mais l'efficacité opérationnelle reste inégale.

Alignements

- **Solution centrée sur la plateforme** : L'évolution des fournisseurs vers des plateformes de sécurité unifiées répond directement au besoin des organisations de consolider la sécurité des applications au sein d'une solution unique.
- **Gestion des SBOMs** : La gestion des SBOMs est devenue un composant central des solutions SCA, en lien direct avec les premières étapes d'adoption de la sécurité des applications par l'organisation.
- **ASPM** : L'accent mis par les fournisseurs sur l'ASPM s'appuie directement sur les premières solutions adoptées par les organisations (SCA, SAST, scan des secrets), l'ASPM étant précisément conçu pour agréger et orchestrer ces solutions fondamentales.
- **Scanners et gestionnaires de secrets** : Les fournisseurs proposent désormais des fonctionnalités avancées en matière de détection et de gestion des secrets, tandis que les organisations intègrent ces solutions dès les premières étapes pour sécuriser leurs actifs critiques.

Écarts

- **Efficacité de la remédiation**: Les solutions d'ASPM introduisent des capacités de réduction du bruit et de priorisation des alertes issues du SAST, DAST et SCA. Toutefois, les organisations continuent de faire face à un volume élevé de faux positifs, ce qui limite l'efficacité de des processus de remédiation.
- **Intégration de l'IA** : Les fonctionnalités d'IA proposées par les fournisseurs restent encore insuffisamment matures pour réduire significativement les faux positifs ou améliorer la priorisation après analyse. Parallèlement, les organisations développent leurs propres solutions internes, mais celles-ci progressent généralement lentement et produisent des résultats encore limités.



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

ANNEXES



METHODOLOGIE D'ANALYSE DE MARCHÉ



Méthodologie de l'étude de marché

Étude de marché

Objectif : Fournir une vision globale du marché européen du S-SDLC et du DevSecOps

Étape 1 - Analyse du marché et tendances structurelles



Identification des solutions de sécurité applicative du marché parmi 12 catégories de solutions définies par l'ANSSI et Wavestone.



Paysage réglementaire : 10 réglementations applicables au DevSecOps, dont 6 Européennes (p. ex. *Cyber Resilience Act*).



Paysage de la recherche : 8 articles, dont 2 européens, traitant du SBOM, DAST^[5], DevSecOps et ML-Ops.

Étape 2 – Focus sur la dynamique de la demande



Nous avons interviewé **9 organisations à travers des ateliers de 1h30, couvrant 7 secteurs** (dont le secteur public, les télécoms, la banque, l'assurance, l'énergie et l'OT*, les CSP**) **comprenant des profils variés** (p. ex. CTO, RSSI, ops, chercheurs).



Pour structurer et standardiser les informations recueillies lors de ces ateliers, nous avons conçu un **questionnaire** basé sur le référentiel de maturité **OWASP SAMM*** - DevSecOps**.

Étape 3 – Focus sur la dynamique de l'offre



Nous avons interviewé **13 fournisseurs** (dont **6 européens**) lors d'**ateliers de 2h**, en concentrant notre analyse sur **3 des 12 catégories de solutions** (SCA/SBOM, ASPM, Secret Managers et Scanners).



Afin de structurer et standardiser les informations issues de ces ateliers, nous avons conçu un **questionnaire de 40 questions** couvrant **6 thématiques clés**.

Les enseignements tirés de l'étude de marché ont été exploités afin de définir des recommandations et des orientations stratégiques à destination de l'écosystème numérique.



Comment analyser le marché?

La méthodologie d'étude est structurée autour de quatre piliers.



Structure du marché

Cartographie mondiale des fournisseurs basée sur des sources publiques, regroupées en trois régions : **États-Unis, Europe et Autres.**

Cartographie complète de l'ensemble des acteurs, **quels que soient leurs tailles (émergents, établis, grands et stratégiques) et leurs niveaux de maturité.**

Classification basée sur les **12 catégories de solutions AppSec**, avec une **segmentation par niveau de maturité et par pertinence stratégique.**

Cette structure constitue une base complète pour analyser la dynamique du marché.



Sélection des catégories de solutions

Parmi les **12 catégories de solutions AppSec** identifiées, trois ont été sélectionnées afin de réaliser une étude plus approfondie :

- **ASPM/Gestion des vulnérabilités**
- **SCA/SBOM**
- **Scanners et gestionnaires de secrets**

La sélection a été guidée par des entretiens menés avec 13 fournisseurs et 9 organisations, une **évaluation de la maturité et des enjeux AppSec**, les **tendances d'adoption** observées ainsi que les **exigences réglementaires.**

Ces catégories constituent le périmètre d'analyse approfondie des solutions.



Périmètre réglementaire

10 réglementations ont été examinées, principalement issues de l'Union européenne (6) et des États-Unis (4).

Le périmètre inclut des exigences relatives à la **transparence logicielle, aux pratiques de sécurité dès la conception (secure-by-design), à la gestion des vulnérabilités** ainsi qu'à la **protection des secrets et identités.**

Parmi les textes analysés, on peut notamment citer : **CRA, NIS2, DORA, EO 14028 et PCI-DSS 4.0.**

Ce périmètre réglementaire constitue une base structurelle pour l'analyse de marché.



Périmètre de la recherche

L'analyse se concentre sur les États-Unis et l'Europe.

La méthodologie repose sur une **revue thématique des publications académiques et industrielles jugées pertinentes pour la sécurité du cycle de vie.**

Une **recherche basée sur des mots-clés (S-SDLC, DevSecOps et termes associés)** a été réalisée, en conservant les publications les plus pertinentes.

Après examen, **8 publications ont été retenues.**

Cette analyse vise à situer les recherches existantes dans le contexte du marché global.



SCÉNARIOS D'ATTAQUE DES PIPELINES CI/CD



Attaques ciblant les pipelines CI/CD

De l'exploitation de vulnérabilités à la compromission du système, les attaques ciblant les pipelines CI/CD démontrent bien qu'il s'agit d'un actif critique à sécuriser

solarwinds



2020

SolarWinds

Une attaque systémique malveillante

2021

Log4Shell

Une vulnérabilité mondiale

2025

Du CI/CD au domaine administrateur

Un chemin d'attaque interne trouvé grâce à un audit

Exemples réels d'attaques CI/CD – SolarWinds

Si votre pipeline CI/CD est compromise, vous devenez l'attaquant de vos propres clients

L'incident SolarWinds montre qu'une seule faille dans la chaîne de build peut silencieusement transformer des mises à jour légitimes en vecteurs d'attaque à grande échelle.

Scénario

Impacts

- SolarWinds de confiance** : 118 000 clients ont téléchargé des mises à jour contenant des malwares. Obstacles à la réparation, pénalités, fuite de données, démissions de cadres. La SEC a accusé SolarWinds et son RSSI (CSO) d'avoir induit les investisseurs en erreur sur la portée de l'incident.
- Clients (clients)** : Agences gouvernementales américaines (Treasury, Commerce, DHS, Department of Energy, etc.), Des entreprises du Fortune 500 (Microsoft, Cisco, Intel, Siemens, etc.), Les fournisseurs de cloud et de cloud hybride ont également été impactés (impression indirecte).
- Globale** : Une mobilisation massive des équipes de réponse à l'incident dans les secteurs public et privé. Reconstitution de systèmes, segmentation des réseaux, isolation des clés et adresses de certificats. Lancement de programmes de remédiation à long terme (insuramment dans des environnements hybrides).

Exemples réels d'attaques CI/CD – Log4Shell

Une vulnérabilité isolée dans une bibliothèque logicielle, dissimulée au sein de ses dépendances, peut déclencher une crise mondiale

Log4Shell a montré que la plupart des organisations avaient peu de visibilité sur les applications et produits utilisant Log4j, y compris les dépendances transitives et les logiciels tiers.

Scénario

Impacts

- Exposition généralisée** : Des dizaines de milliers de produits concernés, aucun secteur ni pays privilégiés. Les organisations ont eu du mal à identifier tous les composants vulnérables (dépendances transitives), à compter ou à prioriser à temps et à définir à quel règlement elles se font.
- Surcharge de la réponse à l'incident** : mobilisation massive des équipes de réponse à l'incident (logique SOC en alerte 24x24 et 3x7). Déploiement de mesures temporaires (par exemple, suppression des fluxes JNDI, règles WAF/IPS). Déploiement de correctifs d'urgence dans un contexte de forte contrainte opérationnelle.
- Perturbations commerciales et opérationnelles** : Dégradation des services clients, interruption d'applications, les réseaux ont temporairement dissipé des fonctionnalités ou bloqué le trafic.

Exemples réels d'attaques CI/CD – basés sur des cas clients

Depuis les pipelines CI/CD, un attaquant peut avoir accès à l'ensemble du système d'information.

Issu d'un cas client observé lors d'un audit, ce scénario démontre qu'une compromission des pipeline CI/CD pouvait servir de point d'entrée à des mouvements latéraux dévoués à une évasion de privilèges jusqu'au niveau administrateur de l'AD.

Scénario

Impacts

Compromission complète du système d'information à partir du pipeline CI/CD

- Perte de confiance dans la chaîne de build** : Tous les builds et versions deviennent suspects. Tout ce qui ne vient pas de build, nécessite de repenser à court et moyen termes, de réinitialiser les secrets, de faire une relation des clés.
- Perturbation de la production et des mises** : Interruptions ou retard entre sur des déploiements. Retard de mise en production en raison d'analyses forensiques nécessaires. Compromission potentielle des environnements clients en cas de réutilisation de code malveillant.
- Atteinte à la réputation et à la marque** : Les Clients peuvent douter de l'intégrité du produit (lié entre au scénario SolarWinds).

Des attaquants ont compromis l'environnement de build de SolarWinds et ont injecté une backdoor dans des mises à jour Orion légitimes, qui ont été distribuées à des milliers de clients.

Une faille critique de type Remote Control Execution dans la bibliothèque de journalisation Log4j a exposé des milliers d'applications qui l'utilisent indirectement via leurs dépendances.

Lors d'un test d'intrusion interne, la redteam a commencé par exploiter l'accès au CI/CD, a profité d'une isolation faible pour récupérer des secrets, et a ensuite pivoté pour compromettre l'Active Directory.



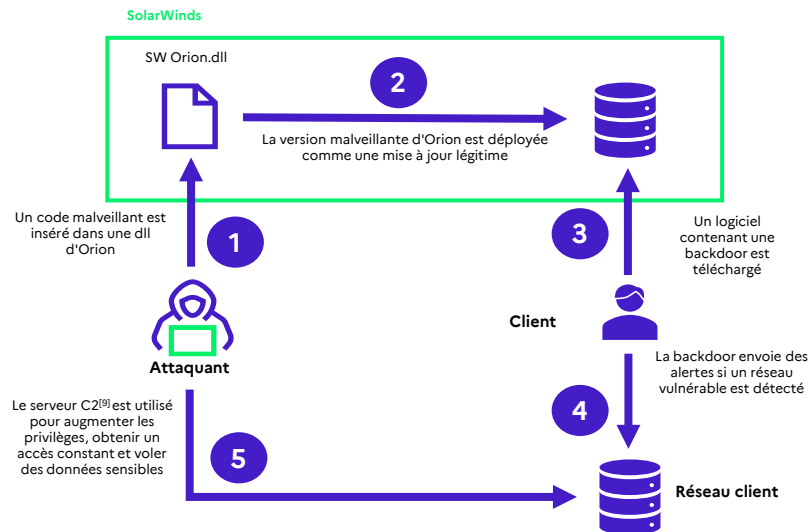
Exemples réels d'attaques CI/CD – SolarWinds

Si votre pipeline CI/CD est compromise, vous devenez l'attaquant de vos propres clients

L'incident SolarWinds montre qu'une seule faille dans la chaîne de *build* peut silencieusement transformer des mises à jour légitimes en vecteurs d'attaque à grande échelle.

Scénario

2020, Source : U.S. GAO



Impacts

- **SolarWinds (le fournisseur)** : ~18 000 clients ont téléchargé des mises à jour contenant des malwares. Dommages à la réputation, procès, chute boursière, démissions de cadres. La SEC a accusé SolarWinds et son RSSI (2023) d'avoir induit les investisseurs en erreur sur la posture cybersécurité.
- **Clients (victimes)** : Agences gouvernementales américaines (Treasury, Commerce, DHS, Department of Energy, etc.). Des entreprises du Fortune 500 (Microsoft, Cisco, Intel, Deloitte, etc.). Les fournisseurs de cloud et de cybersécurité ont également été impactés (exposition indirecte).
- **Ecosystème numérique** : Une mobilisation massive des équipes de réponse à incident dans les secteurs public et privé. Reconstruction de systèmes, segmentation des réseaux, rotation des clés et réémission de certificats. Lancement de programmes de remédiation à long terme (notamment dans des environnements hybrides).



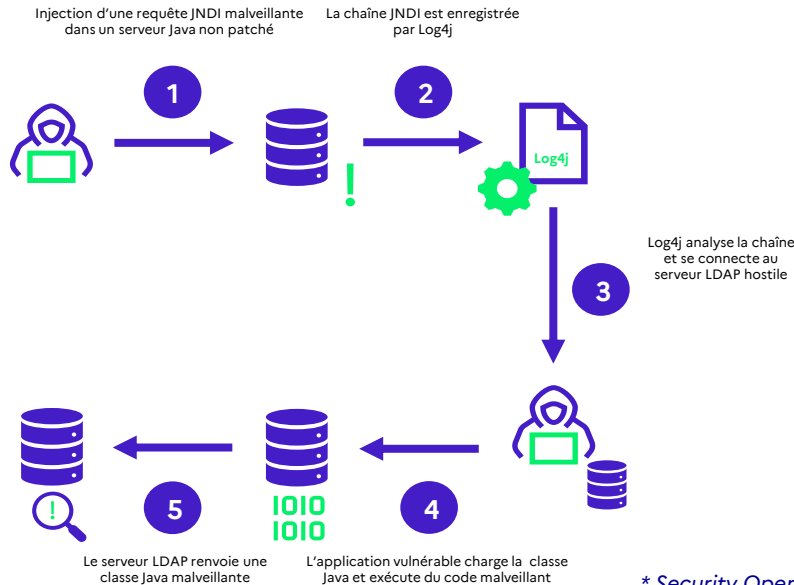
Exemples réels d'attaques CI/CD – Log4Shell

Une vulnérabilité isolée dans une bibliothèque logicielle, dissimulée au sein de ses dépendances, peut déclencher une crise mondiale

Log4Shell a montré que la plupart des organisations avaient peu de visibilité sur les applications et produits utilisant Log4j, y compris les dépendances transitives et les logiciels tiers.

Scénario

CVE-2021-44228, 2021, Source : Splunk



Impacts

- **Exposition généralisée** : Des dizaines de milliers de produits concernés ; aucun secteur n'a été épargné. Les organisations ont eu du mal à identifier tous les composants vulnérables (dépendances transitives), à corriger ou atténuer à temps et à vérifier si une exploitation avait eu lieu.
- **Surcharge de la réponse à incident** : mobilisation massive des équipes de réponse à incident (équipes SOC* en alerte 24h/24 et 7j/7). Déploiement de mesures temporaires (p. ex. suppression des classes JNDI**, règles WAF***). Déploiements de correctifs d'urgence dans un contexte de forte contrainte opérationnelle.
- **Perturbations commerciales et opérationnelles** : Dégradations des services cloud, interruptions d'applications. Les éditeurs ont temporairement désactivé des fonctionnalités ou bloqué le trafic.

* Security Operations Center

** Java Naming and Directory Interface

*** Web Application Firewall

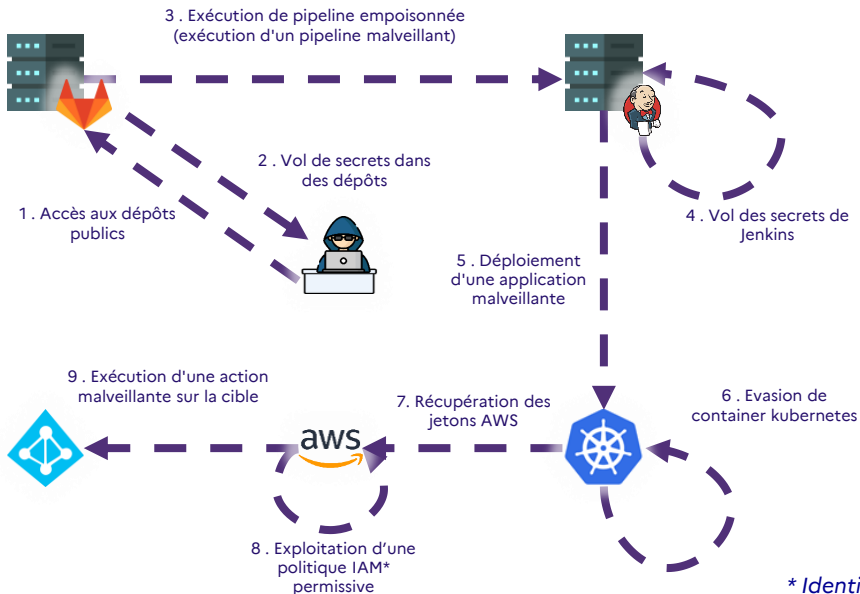


Exemples réels d'attaques CI/CD – basés sur un cas client

Depuis les pipelines CI/CD, un attaquant peut avoir accès à l'ensemble du système d'information.

Issu d'un cas client observé lors d'un audit, ce scénario démontre qu'une compromission des pipeline CI/CD pouvait servir de point d'entrée à des mouvements latéraux aboutissant à une élévation de privilèges jusqu'au niveau administrateur de l'AD.

Scénario



Impacts

Compromission complète du système d'information à partir du pipeline CI/CD

- 1. Perte de confiance dans la chaîne de *build*** : Tous les *builds* et versions deviennent suspects tant qu'ils ne sont pas revalidés. Nécessité de reconstruire à partir de sources saines, de réinitialiser les secrets, de faire une rotation des clés.
- 2. Perturbation de la production et des métiers** : Interruptions ou retour arrière sur des déploiements. Retard de mise en production en raison d'analyses forensiques nécessaires. Compromission potentielle des environnements clients en cas de propagation de code malveillant.
- 3. Atteintes à la réputation et à la marque** : Les clients peuvent douter de l'intégrité du produit (similaire au scénario SolarWinds).

* Identity and Access Management



SYNTHÈSE DES RÈGLEMENTATIONS



Réglementations européennes



DORA 2023

- Évaluer les pratiques et la posture de sécurité des fournisseurs et identifier les dépendances critiques
- Surveillance continue (vulnérabilités, chaîne CI/CD, etc.)
- Offrir une visibilité complète des logiciels tiers et des composants open source utilisés dans leurs produits
- Applicable aux entités financières et aux fournisseurs de services TIC



NIS2 2023

- Implémenter le « Security-By-Design »
- Considérer et corriger les risques associés aux fournisseurs tiers
- Assurer le développement sécurisé du code et l'utilisation des SBOM
- Étendre la gestion des vulnérabilités aux logiciels tiers et open source
- Applicable aux entités moyennes et grandes de l'UE ou offrant des services dans l'UE dans 18 secteurs critiques (y compris : santé, eau, énergie, infrastructures numériques...)



EU Cyber Resilience Act (à venir)

- Implémenter le « Security-By-Design »
- Exiger la transparence concernant l'utilisation de logiciels tiers ou open source, et veiller à ce qu'ils respectent les exigences de sécurité
- Mise à jour via des canaux sécurisés
- Applicable à tous les produits avec éléments numériques disponibles sur le marché européen. Les produits déjà réglementés par des règles sectorielles ne sont pas dans le scope



PLD 2024/25

- Directive générale sur la responsabilité du fait des produits défectueux, également étendue aux produits logiciels
- Précise que les fournisseurs ou développeurs de logiciels peuvent être tenus responsables en cas de faille de sécurité (en cas d'utilisation de produits open source ou tiers mal sécurisés)
- S'applique à tous les produits vendus dans l'UE, y compris les logiciels, leurs mises à jour et les systèmes d'IA, pour les dommages résultant de défauts



LPM 2024/30

- Déclaration obligatoire des incidents et vulnérabilités cybernétiques à l'ANSSI
- Mettre l'accent sur les systèmes « Security-by-Design » et la protection de la chaîne d'approvisionnement logicielle
- Applicable aux Opérateurs d'Importance Vitale (OIV), aux Opérateurs de Services Essentiels (OSE), ainsi qu'aux éditeurs de logiciels ou fournisseurs de services dans des secteurs tels que l'énergie, les transports, la santé et les télécommunications



AI Act

- Exige la mise en œuvre de pratiques de développement sécurisées pour les systèmes d'IA à haut risque, incluant la robustesse, la précision et la résilience face aux attaques
- Favorise les contrôles basés sur le risque, la correction des vulnérabilités et des mécanismes de mise à jour sécurisés, tout au long du cycle de vie du système d'IA
- S'applique à l'ensemble des éditeurs, intégrateurs et distributeurs de systèmes d'IA dans l'UE, avec des obligations plus strictes pour les cas d'usage à haut risque



Réglementations non européennes



U.S. Executive Order 14028

- Exige la mise en œuvre de pratiques de développement sécurisées alignées sur le SSDF du NIST
- Impose la fourniture de SBOM pour les logiciels critiques livrés aux agences fédérales
- Encourage la détection de vulnérabilités, l'architecture zero-trust et la remédiation proactive tout au long du cycle de vie logiciel
- Applicable à tous les éditeurs livrant des logiciels aux agences fédérales américaines



PCI-DSS

- Exige une protection stricte des données des titulaires de carte de paiement dans tous les systèmes et environnements
- Exige des mécanismes robustes d'authentification, de contrôle d'accès et de chiffrement
- Encourage la surveillance continue et les pratiques de développement logiciel sécurisées
- Applicable à toutes les entités qui stockent, traitent ou transmettent des données de cartes de paiement, y compris les commerçants et les prestataires de services



Security Legislation Amendment Critical Infrastructure Protection "SLACIP" Act 2022

- Exige de la part des opérateurs d'infrastructures critiques de mettre en œuvre des programmes complets de gestion des risques cybernétiques, y compris des contrôles sécurisés de développement logiciel
- Encourage l'évaluation des vulnérabilités, la transparence de la *supply chain* et la préparation aux incidents
- Applicable aux secteurs désignés d'infrastructures critiques



Cyber Security Agency – CSA Advisory, Fév. 2025

- Mettre en œuvre la génération de SBOM et automatiser la surveillance en temps réel des vulnérabilités des composants
- Encourager l'utilisation de la CI/CD pour remédier aux risques avant le déploiement
- Applicable aux développeurs logiciels, chaînes d'approvisionnement des logiciels open source



APERÇU DE LA RECHERCHE ACADÉMIQUE



Recherches et publications européennes

Aperçu de la recherche : lectures, périmètre et enseignements

Lectures	Périmètre	Enseignements	Liens
<p>European Cybersecurity Organisation (ECISO) — “Technical Paper: Software Supply Chain Security” (2024)</p>	<p>Analyse de la sécurité de la <i>supply chain</i> logicielle à travers les pratiques modernes de développement, y compris les dépendances, les composants tiers, les solutions, les services et les risques externes de l'écosystème. Il examine également comment ces facteurs impactent l'intégrité logicielle et comment la gestion de la <i>supply chain</i> devrait être intégrée au SDLC.</p>	<ul style="list-style-type: none"> • Mettre en place un S-SDLC structuré intégrant des pratiques de shift-left et des artefacts sécurisés afin de réduire l'exposition aux attaques sur la <i>supply chain</i> logicielle. • La dépendance aux composants tiers implique que la sécurité de la <i>supply chain</i> soit considérée comme une obligation fondamentale du développement, et non comme un simple ajout optionnel. • Fournit des recommandations et des bonnes pratiques. 	<p>ECISO WG6 Technical Paper — Software Supply Chain Security</p>
<p>An Empirical Study of DevSecOps Focused on Continuous Security Testing (2024)</p>	<p>Étude empirique d'un pipeline DevSecOps avec des tests de sécurité continus, axés sur l'intégration précoce de la sécurité SDLC, les tests automatisés et la détection de vulnérabilités</p>	<ul style="list-style-type: none"> • Les équipes de développement doivent être accompagnées, formées et guidées, car les alertes de sécurité sont souvent mal interprétées et les outils actuels génèrent un bruit excessif. • Les scans de sécurité CI/CD doivent rester obligatoires et ne peuvent être désactivés pour des raisons de performance; les pipelines doivent être conçus pour concilier rapidité et sécurité. • Le S-SDLC doit intégrer des indicateurs de sécurité (taux de remédiation des vulnérabilités, couverture des tests, etc.). 	<p>An Empirical Study of DevSecOps Focused on Continuous Security Testing</p>



Cette sélection d'articles, incluant des publications non évaluées par les pairs, demeure sujette à caution et ne reflète que partiellement l'état de l'art en DevSecOps, une analyse plus exhaustive n'ayant pu être menée en raison des contraintes temporelles de l'étude.



Recherches et publications non européennes

Aperçu de la recherche : lectures, périmètre et enseignements

Lectures	Périmètre	Enseignements	Liens
A Reality Check on SBOM-based Vulnerability Management (2025)	Propose une approche de sécurité de la <i>supply chain</i> combinant les SBOMs et l'analyse du code à travers le S-SDLC	<ul style="list-style-type: none"> Les approches traditionnelles de SBOM ne garantissent pas une sécurité exploitable de la <i>supply chain</i> en raison d'un excès de faux positifs. (97,5 % des faux positifs provenant des scanners de vulnérabilités) Les SBOMs doivent être complétés par une analyse basée sur l'utilisation afin de filtrer les vulnérabilités inaccessibles et de réduire le bruit. L'utilisation de l'analyse des appels de fonction (analyse des appels de code réels) pourrait éliminer jusqu'à 63,3 % des faux positifs. 	A Reality Check on SBOM-based Vulnerability Management: An Empirical Study and A Path Forward
Effective Integration of Database Security Tools into SDLC Phases: A Structured Framework (2025)	Propose un framework conceptuel pour intégrer les activités de sécurité des bases de données à chaque phase de développement. Il examine les modèles SDLC courants, décrit les menaces et les types de protections des bases de données, et rapporte des résultats descriptifs des enquêtes fournis par les administrateurs des bases de données.	<ul style="list-style-type: none"> Cette recherche sur le SDLC reste restreinte et centrée sur les bases de données, n'offrant qu'une vision partielle de ce que devrait inclure un SSDLC moderne. Les pratiques de sécurité des bases de données sont intégrées en phases SDLC, avec contrôles d'accès, chiffrement, surveillance, audit, validation de sauvegarde. Les résultats de recherche manquent de rigueur méthodologique et ne fournissent pas de framework validés et généralisables. 	https://www.americaspg.com/article/pdf/3730
Integrating DAST in Kanban and CI/CD (2025)	Propose l'intégration du DAST dans les pipelines DevSecOps : du déploiement en préproduction ou en workflow CI/CD, tests dynamiques automatisés de sécurité des applications en cours d'exécution, à la détection continue des vulnérabilités à l'exécution, en intégrant la sécurité tout au long du cycle de développement	<ul style="list-style-type: none"> Une sécurité efficace nécessite plus que le simple déploiement d'outils ; les équipes doivent être formées, alignées et intégrées dans les objectifs de livraison. Les architectures d'applications modernes introduisent des défis pour l'implémentation de scans, notamment avec le JavaScript dynamique et les protocoles émergents tels que HTTP/3. Les pratiques de développement sécurisées doivent évoluer pour répondre à ces contraintes techniques. 	Integrating DAST in Kanban and CI/CD: A Real-World Security Case Study



Cette sélection d'articles, incluant des publications non évaluées par les pairs, demeure sujette à caution et ne reflète que partiellement l'état de l'art en DevSecOps, une analyse plus exhaustive n'ayant pu être menée en raison des contraintes temporelles de l'étude.



Recherches et publications non européennes

Aperçu de la recherche : lectures, périmètre et enseignements

Lectures	Périmètre	Enseignements	Liens
<p>A Practical Guide for Building Robust AI/ML Pipeline Security (2025) & DevSecMLOps: A Security Framework for Machine Learning (2025)</p>	<ul style="list-style-type: none"> Propose un cycle de vie (MLSecOps) qui intègre les contrôles de sécurité, l'orchestration de la chaîne d'outils, la protection de la <i>supply chain</i>, la surveillance continue et le « policy-as-code » pour les pipelines IA/ML. Propose un framework « DevSecMLOps » : une architecture structurée « security-by-design » pour les pipelines ML 	<ul style="list-style-type: none"> Les pratiques traditionnelles des SSDLC ne couvrent pas les risques introduits par les pipelines, modèles et flux de données ML/IA. MLSecOps étend le DevSecOps aux systèmes ML/IA, les données, modèles et pipeline sont des composants critiques de la <i>supply chain</i> dont la sécurité doit être intégrée de bout en bout. La sécurité d'un système ML/IA ne doit pas être ajoutée à la dernière minute, mais intégrée de manière globale, continue et automatique dans tout le pipeline. 	<p>SSF_MLSecOps_Whitepaper.pdf</p> <p>DevSecMLOps: A Security Framework for Machine Learning Pipelines</p>
<p>Software security in practice: knowledge and motivation (2025)</p>	<p>Propose une étude qualitative explorant comment les développeurs acquièrent des connaissances en sécurité et ce qui les motive à appliquer des pratiques de codage sécurisé. Introduit une taxonomie des modes d'apprentissage et applique la théorie de l'autodétermination pour évaluer les styles de motivation.</p>	<ul style="list-style-type: none"> Les développeurs privilégient l'apprentissage contextuel à des formations génériques. La motivation intrinsèque conduit à une adoption plus forte de la sécurité que les pratiques motivées par un besoin de conformité. Parmi les obstacles à l'adoption on peut citer : un mauvais ajustement des outils, un manque de soutien et une culture de sécurité faible. Recommande l'apprentissage par les pairs, des exercices basés sur des scénarios et la documentation intégrée. 	<p>https://academic.oup.com/cybersecurity/article/11/1/tyaf005/8071721</p>



Cette sélection d'articles, incluant des publications non évaluées par les pairs, demeure sujette à caution et ne reflète que partiellement l'état de l'art en DevSecOps, une analyse plus exhaustive n'ayant pu être menée en raison des contraintes temporelles de l'étude.



SÉLECTION DES CATÉGORIES STRATÉGIQUES



Focus – Catégories stratégiques – Echelle détaillée

HISTORIQUE DE SÉCURITÉ

Score 1 – Faible

- Principalement mentionné dans des recommandations théoriques, rarement documenté dans des analyses post-mortem concrètes. Il apparaît rarement dans les retours d'expérience ou dans les guides opérationnels mis à jour après des crises majeures.

Score 2 – Moyen

- Plusieurs incidents où la catégorie a contribué partiellement : utilisée sur certains systèmes / projets, ou pour affiner l'analyse d'impact ou la remédiation, sans être centrale dans la gestion de crise.

Score 3 – Élevé

- À plusieurs reprises au centre des incidents majeurs, clairement identifiés comme un facteur clé de réussite : p. ex. les SBOMs pour inventorier l'exposition à Log4Shell, les SAST utilisés pour une revue de code à grande échelle après une faille de sécurité, les questionnaires de vulnérabilités pour piloter des campagnes de patch massives.

MENTIONNÉ DANS DES RÉFÉRENTIELS

Score 1 – Faible

- Mentionné dans 0 à 2 référentiels majeurs
- Lorsqu'il est mentionné, il apparaît uniquement comme un sous-contrôle ou un exemple, et non comme une exigence autonome

Score 2 – Moyen

- Référencé explicitement dans 3 à 5 référentiels majeurs
- Souvent présenté comme une pratique recommandée ou un contrôle, parfois intégré dans des exigences plus larges

Score 3 – Élevé

- Contrôle explicite et récurrent dans >5 référentiels majeurs
- Dispose souvent de son **propre contrôle, chapitre ou exigence** (p. ex. « SBOM », « Gestion des secrets », « gestion des vulnérabilités »)

CONNAISSANCE DU SUJET

Score 1 – Faible

- Sujet presque jamais abordé spontanément par les organisations ; principalement introduit par des consultants ou fournisseurs
- Rarement présent dans les présentations exécutives ou les plans directeurs de sécurité. Très peu de conférences ou notes d'analystes dédiées à cette catégorie

Score 2 – Moyen

- Sujet régulièrement abordé dans des comités de pilotages, des conseils d'architecture, etc.
- Présent dans une partie non négligeable des appels d'offre ou des plans de pilotages de sécurité, mais pas encore systématique

Score 3 – Élevé

- Largement connu des RSSI et des DSI, souvent cité parmi les 3 priorités principales lors des discussions
- Fréquemment présent dans les appels d'offres, les stratégies et les dialogues réglementaires

ADOPTION PAR LES ORGANISATIONS

Score 1 – Faible

- La majorité des organisations sont uniquement à l'étape d'idée, d'étude ou de veille
- Quelques PoCs isolés, souvent arrêtés ou non industrialisés

Score 2 – Moyen

- Une part significative des organisations réalise des PoC ou des déploiements sur périmètre limité
- Les solutions sont parfois intégrées dans les processus CI/CD ou de sécurité, mais la couverture reste partielle

Score 3 – Élevé

- Cette catégorie est déployée en production à grande échelle pour de nombreuses organisations (plusieurs applications / périmètres / groupe entier)
- Les processus et l'organisation sont en place (rôles, responsabilité, budget récurrent, indicateurs de performance). La catégorie apparaît souvent comme une exigence standard dans les appels d'offres et les directives internes



Focus – Catégories stratégiques – Catégories sélectionnées

Parmi les 12 catégories de solutions, trois ont été sélectionnées pour mener des entretiens avec les fournisseurs

Catégorie	Enjeux	Acteurs européens	Niveau de maturité
SCA / SBOM	Fort – Régulation, sujet post-quantique à venir, sujet IA. <i>Mentionné dans des référentiels : 3</i> <i>Historique de sécurité : 3</i>	Oui	Haute – Sensibilisation (SolarWinds, Log4shell, réglementations : DORA, etc.). <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 3</i>
Gestionnaires et scanners de secrets	Fort – Le Top 10 de l'OWASP comprend de nombreux risques liés à l'identité. <i>Cité dans des référentiels : 3</i> <i>Historique de sécurité : 3</i>	Oui	Moyenne – Sujet observé chez certains clients de Wavestone. Exemple récent de manque de maturité : la base de données McDonald's (mot de passe par défaut 123456). <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 2</i>
ASPM	Fort – Risques associés à des vulnérabilités connues des attaquants, avec des chemins d'attaque identifiés et maîtrisés par des solutions offensives (CVE, etc.). <i>Mentionné dans des référentiels : 3</i> <i>Historique de sécurité : 3</i>	Oui	Moyenne – Gouvernance hétérogène observée parmi les clients de Wavestone. Les efforts restent principalement concentrés sur la remédiation en production. <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 2</i>

Échelle - Enjeux

Mentionné dans des référentiels	3	Low	Moyen	Fort
	2	Low	Moyen	Moyen
	1	Low	Low	Low
		1	2	3

Historique de sécurité

Échelle – Niveau de maturité

Connaissance du sujet	3	Low	Modéré	Haut
	2	Low	Modéré	Modéré
	1	Low	Low	Low
		1	2	3

Adoption par les organisations



Focus – Catégories stratégiques – Catégories non sélectionnées

Catégorie	Enjeux	Acteurs européens	Niveau de maturité
SAST	Fort – Élément de base de la sécurité applicative <i>Mentionné dans des référentiels : 3</i> <i>Historique de sécurité :3</i>	Oui	Haute – Fonctionnalité de sécurité de base observée chez les clients de Wavestone <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 3</i>
IaC Scanner	Fort – L'expansion du Cloud entraîne une augmentation des risques liés aux erreurs de configuration <i>Mentionné dans des référentiels : 3</i> <i>Historique de sécurité :3</i>	Aucun	Moyenne – Adoption croissante de l'IaC, un marché en évolution <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 2</i>
Scanner d'artefacts applicatifs	Moyen – Cité dans les bonnes pratiques CI/CD <i>Mentionné dans des référentiels : 2</i> <i>Historique de sécurité :2</i>	Aucun	Moyenne – Sujet hétérogène, clairement identifié pour les conteneurs, moins mature pour les packages et binaires <i>Connaissance du sujet : 2</i> <i>Adoption par les organisations : 2</i>
Solutions de modélisation des menaces	Moyen – Intervient le plus tôt possible dans la chaîne de développement, constituant une base de sécurité pour les applications et le code source <i>Mentionné dans des référentiels : 2</i> <i>Historique de sécurité :3</i>	Oui	Moyenne – Adoption limitée aux organisations matures. Les méthodes manuelles restent prédominantes. <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 2</i>

Échelle - Enjeux

Mentionné dans des référentiels	3	Low	Moyen	Fort
	2	Low	Moyen	Moyen
	1	Low	Low	Low
		1	2	3

Historique de sécurité

Échelle – Niveau de maturité

Connaissance du sujet	3	Low	Modéré	Haut
	2	Low	Modéré	Modéré
	1	Low	Low	Low
		1	2	3

Adoption par les organisations



Focus – Catégories stratégiques – Catégories non sélectionnées

Catégorie	Enjeux	Acteurs européens	Niveau de maturité
DAST	Moyen – Permet la détection de vulnérabilités exploitables durant l'exécution, avant la mise en production <i>Mentionné dans des référentiels : 2</i> <i>Historique de sécurité :2</i>	Aucun	Modéré – Automatisation complexe via CI/CD (temps d'exécution long, consommation importante de ressources si des tests pertinents sont nécessaires, etc.) <i>Connaissance du sujet : 3</i> <i>Adoption par les organisations : 2</i>
Solutions de signature d'artefacts	Faible – Thématique classée en bas de la liste du TOP 10 OWASP (CICD-SEC-9 : contrôle de l'intégrité des artefacts insuffisant) <i>Mentionné dans des référentiels : 2</i> <i>Historique de sécurité :1</i>	Aucun	Faible – L'adoption et l'ergonomie des outils demeurent inférieures à la maturité atteinte par des solutions comme le SAST et le SCA. <i>Connaissance du sujet : 2</i> <i>Adoption par les organisations : 1</i>
RASP	Faible – La maturité globale du S-SDLC nécessite de se concentrer sur la maîtrise des fondamentaux (SAST, SBOM, gestion des vulnérabilités, etc.) avant d'envisager des solutions avancées telles que le RASP. <i>Mentionné dans des référentiels : 1</i> <i>Historique de sécurité :1</i>	Aucun	Faible – Techniquement prometteur, mais sa complexité, son faible taux d'adoption et ses problèmes de performance freinent son adoption par le grand public. <i>Connaissance du sujet : 2</i> <i>Adoption par les organisations : 1</i>

Échelle - Enjeux

Mentionné dans des référentiels	3	Low	Moyen	Fort
	2	Low	Moyen	Moyen
	1	Low	Low	Low
		1	2	3

Historique de sécurité

Échelle – Niveau de maturité

Connaissance du sujet	3	Low	Modéré	Haut
	2	Low	Modéré	Modéré
	1	Low	Low	Low
		1	2	3

Adoption par les organisations



S-SDLC / DEVSECOPS - QUELS SONT LES DÉFIS ACTUELS ET FUTURS DU MARCHÉ EUROPÉEN ?

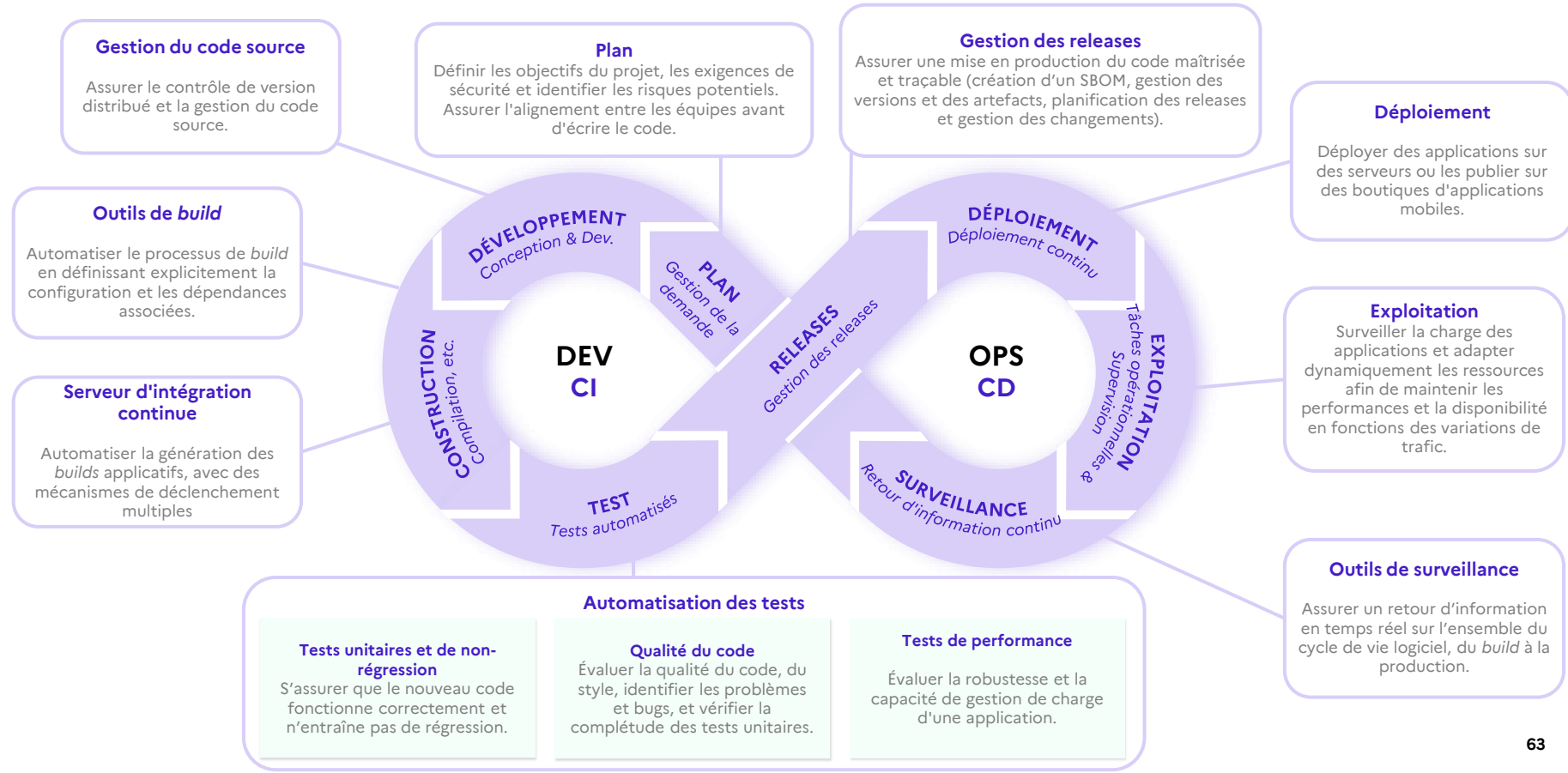
OUTILS



PIPELINES CI/CD – ACTIVITÉS & COMPOSANTS CLÉS



Étapes et activités du pipeline CI/CD





Composants CI/CD – Gestionnaire de code source

Gestionnaire de code source

Open source : [GitLab CE](#), [Gitea](#), [Gogs](#), [Forgejo](#), [Codeberg](#)...

Commerciaux : [GitHub](#), [GitLab EE](#), [Bitbucket](#), [Tuleap](#), [Tuleap Azure DevOps Repos](#), [AWS CodeCommit](#), [Google Cloud Source repositories](#)...

Héberge le **code source**,
contrôle les versions et
assure le développement
collaboratif .

Pourquoi c'est important : il est à la base de la confiance. Toute compromission se répercute en aval de la chaîne.

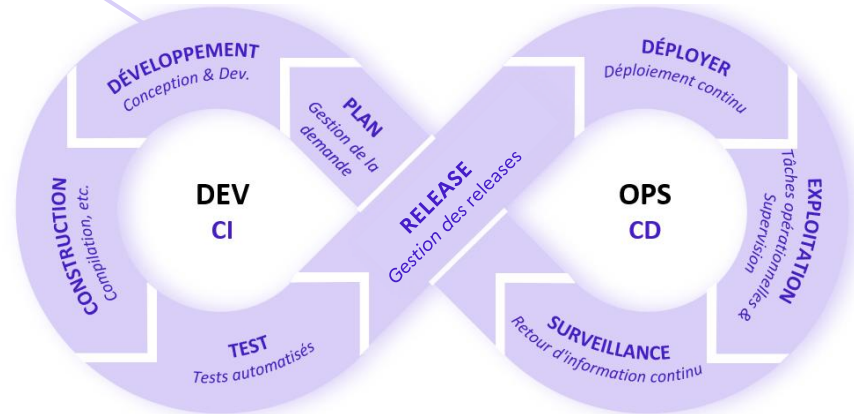
Impact de l'attaque : altération du code source, vol de propriété intellectuelle ou injection de logiciels malveillants dès le développement du code.

Exemple d'attaque : un attaquant compromet le token d'accès personnel GitHub d'un développeur associé à une backdoor dans une bibliothèque interne largement utilisée

Les gestionnaires de code source font partie de la phase de développement

Gestion du code source

Assurer le contrôle de version distribué et la gestion du code source.





Composants critiques CI/CD – Orchestrateur à intégration continue

Intégration continue / Déploiement continu

Open source : [Jenkins](#), [GitLab CE](#), [Spinnaker](#), ...

Commerciaux : [Github actions](#), [Bitbucket Pipelines](#), [Travis CI](#), [Azure Pipelines](#), [AWS CodePipeline](#), [Google Cloud Build](#)...

Orchestre une intégration de code fiable et rapide, déclenche **des compilations** et **des tests** automatisés.

Pourquoi c'est important : les outils CI fonctionnent avec des privilèges élevés, souvent avec des jetons d'accès privilégiés.

Impact de l'attaque : contrôle total des flux de compilation et de test, injection d'artefacts malveillants, contournement des contrôles de sécurité.

Exemple d'attaque : un plugin Jenkins vulnérable permet l'exécution de code à distance (RCE^[8]) sur le serveur master Jenkins. L'attaquant peut alors récupérer tous les identifiants stockés

L'orchestrateur d'intégration continue fait partie des phases de développement et de *build* (construction)



[8] Remote Code Execution



Composants critiques CI/CD – Gestionnaire d'artefacts

Gestionnaire d'artefacts

Open source : [GitLab Package Registry \(CE\)](#), [docker/container registry](#), [JFrog Artifactory OSS](#), [Harbor](#), ...

Commerciaux : [GitHub Packages](#), [JFrog Artifactory](#), [Sonatype Nexus Repository](#), [Azure Artifacts](#), [AWS CodeArtifact](#), [Google Artifact Registry](#)...

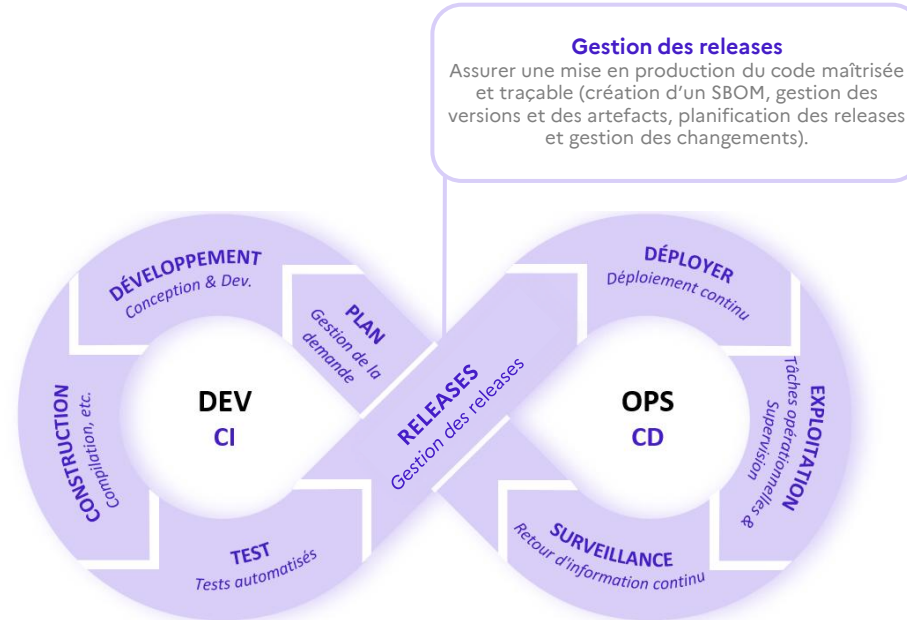
Stocke et gère les **artefacts** (binaires compilés, images de conteneurs, etc.), garantissant la **gestion des versions** sur tout le pipeline.

Pourquoi c'est important : toute personne disposant d'un accès en écriture aux dépôts d'artefacts peut insérer d'autres artefacts malveillants directement dans le flux de livraison.

Impact de l'attaque : propagation de *builds* malveillants, compromission de la *supply chain*.

Exemple d'attaque : un attaquant accède au dépôt interne Maven et remplace le fichier *payment.jar* par une version malveillante, permettant ainsi l'exfiltration de données de carte bancaire.

Le gestionnaire d'artefacts fait partie de la phase de release





Composants critiques CI/CD – Outil de Provisionnement

Outil de Provisionnement

Open source : [Ansible](#), [Open Source Puppet](#), [OpenVox](#), [OpenTofu](#), [Chef](#), [pyinfra](#)...

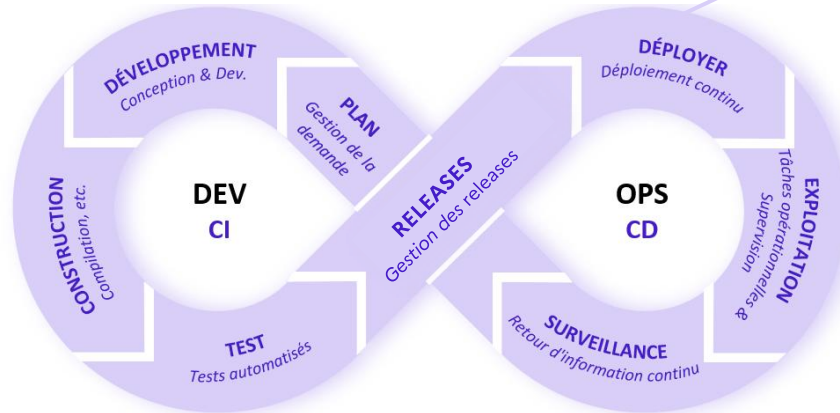
Commerciaux : [Terraform Enterprise](#), [perforce Puppet](#), [AWS CloudFormation](#), [Azure Resource Manager](#)...

Automatise la création et la configuration de l'infrastructure dans **des processus de déploiement reproductibles**.

Pourquoi c'est important : Les outils de provisionnement codifient l'infrastructure et stockent des identifiants disposant de privilèges étendus.
Impact des attaques : altération de l'infrastructure, création de serveurs malveillants ou collecte de secrets.

Exemple d'attaque : un token API Terraform Cloud stocké sur l'ordinateur d'un ingénieur est volé. L'attaquant l'utilise pour modifier le code et l'état de l'IaC.

L'outil de provisionnement fait partie du déploiement



Déploiement

Déployer des applications sur des serveurs ou les publier sur des boutiques mobiles d'applications.



Composants critiques CI/CD – Conteneurisation et orchestration

Conteneurisation et orchestration

Open source : [docker](#), [podman](#), [AppTainer](#), [Linux Container \(Incus, LXC...\)](#), [Kubernetes](#), [docker swarm...](#)

Commerciaux : [RedHat OpenShift](#), [Amazon Elastic Kubernetes Service](#), [Azure Kubernetes Service](#), [Google Kubernetes Engine](#), [Suse Rancher...](#)

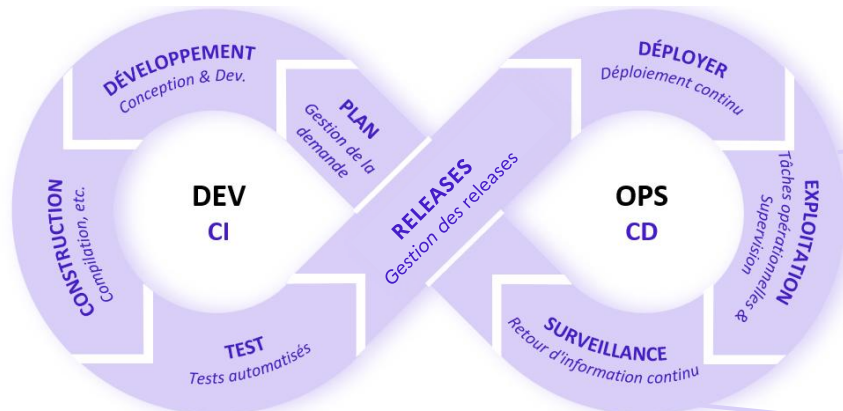
Gère les charges de travail et services des conteneurs, en optimisant les opérations de déploiement et d'exécution.

Pourquoi c'est important : ces outils gèrent les secrets, la mise à l'échelle et la communication entre services. Une compromission à ce niveau permet une prise de contrôle opérationnel du système.

Impact de l'attaque : manipulation de conteneurs en production, interception du trafic, vol de secrets.

Exemple d'attaque : Un pod s'exécute avec un compte de service Kubernetes sur-privilegié. L'attaquant utilise ce token pour déployer un mineur de crypto.

La conteneurisation et l'orchestration font partie de l'exploitation et de la surveillance



Exploitation

Surveiller la charge des applications et adapter dynamiquement les ressources afin de maintenir les performances et la disponibilité en fonctions des variations de trafic.

Outils de surveillance

Assurer un retour d'information en temps réel sur l'ensemble du cycle de vie logiciel, du *build* à la production.



Directives de sécurité de haut niveau CI/CD



Les lignes directrices suivantes constituent **des recommandations de haut niveau** pour renforcer la **sécurité et la résilience de la chaîne d'outils CI/CD**. En abordant **les aspects fondamentaux de sécurité**, ces directives visent à protéger la chaîne d'outils contre les menaces précédemment présentées, garantissant ainsi **une protection renforcée** à travers toute la chaîne d'outils.

Évaluation des risques des composants de la chaîne CI/CD

- Effectuer une évaluation des risques des composants de la chaîne d'outils CI/CD (p. ex. Analyse de risque, modélisation des menaces) avant leur déploiement et leur utilisation.

Gestion de l'identité et des accès

- Principe du moindre privilège et contrôle d'accès basé sur les rôles
- Authentification multi-facteur, politique de connexion unique et de verrouillage des comptes
- Intégration sécurisée des API* avec les composants CI/CD
- Gestion du cycle de vie des comptes
- Robustesse et gestion des identifiants

Surveillance et détection

- Journalisation de sécurité
- Surveillance et détection d'incidents

Assurance sécurité

- Audits réguliers des composants CI/CD

Gestion des infrastructures, des réseaux et des systèmes

- Sécuriser l'infrastructure et le réseau sous-jacents
- Communications sécurisées entre les composants CI/CD
- Effectuer les mises à jour de sécurité et la gestion des correctifs

Protection de l'intégrité des artefacts

- Artefacts et dépendances fiables et sécurisés
- Centralisation et gestion du versionnement des artefacts
- Vérifications d'intégrité des artefacts et de leur signature
- Chiffrement des artefacts

Continuité et reprise

- Gestion des sauvegardes
- Plan de reprise après sinistre

*Analyse de risque des principaux composants
d'un pipeline CI/CD, accompagnée de
recommandations détaillées*



Voir analyse
des risques

* Application Programming Interface



PIPELINES CI/CD – ANALYSE DE RISQUES ET RECOMMANDATIONS



Méthodologie de l'analyse de risque

→ OBJECTIFS


Cette étude vise à atteindre les objectifs suivants :

 **Identifier les risques et vulnérabilités** inhérents à la chaîne d'outils CI/CD

 **Aligner les risques** identifiés avec les références MITRE ATT&CK pertinentes

 **Catégoriser les vulnérabilités** à l'aide du **TOP 10 CI/CD OWASP** et des exemples de **CWE** et **CVE**

Les scores CVSS associés à chaque CVE sont fournis. Il est pris en compte dans l'évaluation d'impact du risque mais cette évaluation est principalement qualitative.

 **Proposer des mesures de sécurité** pour atténuer les risques identifiés et **renforcer la posture globale de sécurité** du pipeline CI/CD

Échelle d'impact

Niveau	Description
1 (Très faible impact)	Perturbation minimale du pipeline CI/CD Perte ou exposition de données insignifiante Aucun impact notable sur les opérations métier Récupération aisée, sans coût ni effort significatif
2 (Faible impact)	Perturbation limitée du pipeline CI/CD Perte ou exposition limitée de données Impact mineur sur les opérations métier, entraînant de légères difficultés Récupération avec un coût et un effort faibles
3 (Impact modéré)	Perturbation modérée du pipeline CI/CD Perte ou exposition de données non critiques Impact notable sur les opérations métier, retardant les projets Nécessite un effort et un coût modérés pour la récupération
4 (Impact fort)	Perturbation grave du pipeline CI/CD Perte ou exposition de données critiques Impact majeur sur les opérations métier, pouvant entraîner l'arrêt des projets Nécessite un effort et un coût substantiels pour la récupération, impliquant potentiellement un support externe

Échelle de vraisemblance

Niveau	Description
1 (Très improbable)	La source de menace a très peu de chances d'atteindre son objectif en suivant l'un des modes opératoires considérés. La vraisemblance du scénario de risque est très faible.
2 (Peu probable)	La source de menace a relativement peu de chances d'atteindre son objectif en suivant l'un des modes opératoires considérés. La vraisemblance du scénario de risque est faible.
3 (Probable)	La source de menace parviendra probablement à atteindre son objectif en suivant l'un des modes opératoires considérés. La vraisemblance du scénario de risque est élevée.
4 (Très probable)	La source de menace atteindra probablement son objectif en suivant l'un des modes de fonctionnement envisagés. La probabilité du scénario risqué est élevée.

Échelle de complexité de la mise en œuvre

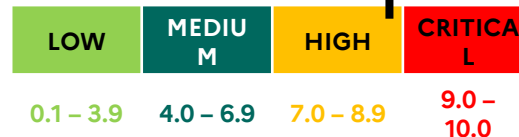
Niveau	Description
+	Modifications minimales, mise en œuvre rapide et aisée, faibles exigences en termes de coût et de ressources.
++	Nécessite des modifications modérées, une planification et une coordination préalables, implique une allocation de coûts et de ressources limitée.
+++	Modifications substantielles, planification approfondie, exigences élevées en termes de coût et de ressources, engagement sur le long terme.



Méthodologie de l'analyse de risque

RISQUES IDENTIFIÉS (EXEMPLE)

ID	Description du risque	Impact	Probabilité	Référence Mitre Att&ck	Vulnérabilités liées au risque (OWASP Top 10 CI/CD ; CWE & CVE)
R05	Attaque par déni de service (DoS) sur les instances SCM entraînant une perturbation du service			T1499 (Endpoint Denial of Service)	OWASP: Insufficient Flow Control Mechanisms CWE-770 (Allocation of Resources Without Limits or Throttling) CVE-2024-9631 (only for Gitlab) CVSS: 7,5



RECOMMANDATIONS (EXEMPLE)

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLSA**
REC07	R05	Réseau	Mettre en œuvre des mesures de sécurité réseau telles que la limitation de fréquence et les services de protection DDoS pour protéger les serveurs SCM	+++	Level 2 – Operational management	Level 3 – deployment

Le niveau de maturité SAMM est évalué selon le *framework* SAMM* en utilisant une approche en trois étapes et les principes du framework

Le niveau de sécurité SLSA est évalué selon le *framework* SLSA* en utilisant une approche en deux étapes et les principes du framework

* Software Assurance Maturity Model

** Supply-chain Levels for Software Artifacts



Risques globaux CI/CD



Les chaînes d'outils CI/CD sont exposées à divers risques de sécurité susceptibles de compromettre l'intégrité, la disponibilité et la confidentialité des logiciels. Les principales vulnérabilités incluent des **contrôles d'accès insuffisants**, des **dépendances non vérifiées**, une **supervision inadéquate** et des **plans de reprise insuffisamment robustes**. L'identification et le traitement de ces risques sont essentiels au maintien d'un environnement de développement sécurisé.

Thème	Description du risque	Impact	Probabilité	Vulnérabilités liées au risque (OWASP Top 10 CI/CD)
Infrastructure et segmentation	Absence de segmentation ou d'isolation entre les composants de la chaîne d'outils — compromission d'un composant pouvant se propager à l'ensemble de la chaîne.			OWASP: Insufficient Flow Control Mechanisms
Contrôle d'accès	Gestion insuffisante des identifiants, politiques RBAC mal configurées, gestion des identités et des accès inadéquate — absence de contrôle des flux, accès non autorisés et risques d'élévation de privilèges.			OWASP: Inadequate Identity and Access Management OWASP: Insufficient PBAC (Pipeline-Based Access Controls) OWASP: Insufficient Credential Hygiene
Sécurité des artefacts	Altération des artefacts et validation d'intégrité insuffisante — risques liés à un chiffrement ou à des contrôles d'intégrité inadéquats, pouvant conduire à des artefacts compromis.			OWASP: Improper Artifact Integrity Validation OWASP: Poisoned Pipeline Execution (PPE)
Exécution des artefacts	Exécution d'artefacts non fiables issus de dépôts non maîtrisés ou malveillants et exécution de pipelines empoisonnés — risque de déploiements compromis via des sources non fiables.			OWASP: Improper Artifact Integrity Validation OWASP: Poisoned Pipeline Execution (PPE)
Dépendances	Utilisation de dépendances externes non fiables ou non vérifiées — risques liés à des bibliothèques open source ou des outils tiers ne faisant pas l'objet d'une validation ou d'une gouvernance appropriée.			OWASP: Dependency Chain Abuse OWASP: Ungoverned Usage of 3rd Party Services





Risques globaux CI/CD

Thème	Description du risque	Impact	Probabilité	Vulnérabilités liées au risque (OWASP Top 10 CI/CD)
Journalisation et surveillance	Supervision, journalisation et visibilité insuffisantes — incidents de sécurité non détectés et réponse aux incidents retardée en raison d'une intégration inadéquate des journaux avec les capacités de détection (alertes natives, SIEM...).			OWASP: Insufficient Logging and Visibility
Incident et restauration	Plans de reprise et outils de détection inadéquats — temps d'arrêt significatifs et déploiements compromis dus à des processus de sauvegarde insuffisamment maîtrisés.			/
Protection des données	Communications non sécurisées entre les composants de la chaîne d'outils — interception ou manipulation des données via des canaux non sécurisés.			OWASP: Insecure System Configuration
Configuration du système	Gestion des correctifs inadéquate et configuration système non sécurisée — systèmes exposés aux vulnérabilités résultant à la fois de mauvaises configurations et de correctifs non appliqués.			OWASP: Insecure System Configuration
Contrôles de flux	Mécanismes de contrôle insuffisants et absence de provisionnement automatisé — déploiements non autorisés ou indésirables en environnement de production, augmentant le risque d'introduction de vulnérabilités, de mauvaises configurations ou de code instable dans des systèmes critiques.			OWASP: Insufficient Flow Control Mechanisms





Principes directeurs de sécurité des pipelines CI/CD



Les recommandations de durcissement suivantes proposent des préconisations de haut niveau visant à **renforcer la sécurité et la résilience de la chaîne d'outils CI/CD**. En couvrant **les aspects fondamentaux de la sécurité**, ces recommandations visent à **protéger la chaîne d'outils contre les risques précédemment identifiés**, en garantissant des garde-fous renforcés sur l'ensemble de la chaîne.

Évaluation des risques de la chaîne d'outils CI/CD

CICD-RIA-01 : Évaluation des risques des composants de la chaîne d'outils CI/CD

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Effectuer une évaluation des risques des composants de la chaîne d'outils CI/CD (p. ex. analyse de risque, modélisation des menaces) au moins avant le déploiement et l'utilisation.

Gestion des infrastructures, des réseaux et des systèmes

CICD-IFN-01 : Sécuriser l'infrastructure et le réseau sous-jacents

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- S'assurer que l'infrastructure et le réseau (p. ex. serveurs, systèmes d'exploitation) sur lesquels reposent les composants CI/CD sont correctement sécurisés.
- Restreindre l'accès des composants CI/CD aux équipements de l'organisation.

CICD-IFN-02 : Communications sécurisées entre composants CI/CD

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Imposer des protocoles sécurisés pour les communications impliquant des composants CI/CD (p. ex. HTTPS).
- Utiliser des certificats à jour émis par la PKI de l'organisation.

CICD-IFN-03 : Mises à jour de sécurité et gestion des correctifs

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Utiliser la dernière version des composants CI/CD et des plugins (logiciels) et appliquer les correctifs de sécurité dès que possible.



Principes directeurs de sécurité des pipelines CI/CD

Gestion de l'identité et des accès (1/2)

CICD-IAM-01 : Principe du moindre privilège et contrôle d'accès basé sur le rôle

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Mettre en œuvre un modèle strict de contrôle d'accès basé sur les rôles (RBAC), respectant les principes du moindre privilège et nécessaire.
- Limiter le nombre de comptes privilégiés et assurer une séparation appropriée des tâches.
- Recertifier tous les accès périodiquement (au moins chaque année) et corriger les comptes inutilisés.

CICD-IAM-02 : Authentification multifactorielle, politique de connexion unique et de verrouillage

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Activer le Single Sign-On (SSO) et l'authentification multifacteur (MFA) sur les composants CI/CD.
De préférence utiliser WebSSO pour s'appuyer sur un système d'authentification centralisé ; Si le SSO n'est pas activé, exiger la MFA pour tous les utilisateurs.
- Configurer les politiques de verrouillage des comptes pour limiter les attaques de brute-force autant que possible.
- Activer l'accès conditionnel lorsque cela est possible, pour une posture de sécurité Zero-Trust améliorée (*optionnel*).

CICD-IAM-03 : Intégration API sécurisée avec les composants CI/CD

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Imposer des mécanismes d'authentification et d'autorisation robustes (p. ex. jetons OAuth2/OIDC) pour une intégration sécurisée des API avec les composants CI/CD.
- Renouveler régulièrement les clés API et les jetons.

CICD-IAM-04 : Gestion du cycle de vie des comptes

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- S'assurer que le cycle de vie du compte est géré correctement et de manière sécurisée.
- Mettre en place un processus de gestion du provisionnement et du déprovisionnement des utilisateurs pour les composants CI/CD (p. ex. nouvelles arrivées, départs).
- Supprimer les comptes intégrés avec des identifiants par défaut.



Principes directeurs de sécurité des pipelines CI/CD

Gestion de l'identité et des accès (2/2)

CICD-IAM-05 : Gestion et robustesse des identifiants et secrets

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Imposer des mots de passe robustes et complexes et assurer une rotation régulière des mots de passe / secrets.
- Stocker les identifiants et secrets (y compris les clés SSH, mots de passe, jetons OAuth/OIDC, clés API) dans un coffre-fort et s'assurer qu'ils ne soient jamais stockés en clair et/ou exposés (p.ex. dans le code source, les dépôts, les logs ou les messages d'erreur).

Protection de l'intégrité des artefacts (1/2)

CICD-AIP-01 : Artefacts et dépendances fiables et sécurisés

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Utiliser des artefacts et dépendances (p. ex. images de base de conteneurs, bibliothèques logicielles) de confiance pour garantir la sécurité, la stabilité et la conformité. Éviter d'utiliser des artefacts provenant de registres publics et de dépôts sans une vérification de sécurité appropriée (incluant l'analyse des vulnérabilités, l'évaluation de la sécurité)
- Générer un SBOM (Software Bill Of Materials) pour lister tous les artefacts, dépendances et versions associées sur lesquels s'appuient les produits et la chaîne CI/CD
- Verrouiller les versions des dépendances logicielles afin de garantir la cohérence des environnements et de prévenir d'éventuels effets de bord (p. ex. interdire l'usage du tag « latest »).

CICD-AIP-02 : Centralisation et versionnement des artefacts

Applicable pour : Gestion du code source (SCM), registres d'artefacts.

- Utiliser des solutions de gestion du code source et des registres d'artefacts pour centraliser, suivre et versionner les artefacts (p. ex. code source, images de conteneurs, modules Infrastructure-as-Code, scripts de configuration).
- Utiliser des dépôts et registres privés pour stocker des artefacts confidentiels.



Principes directeurs de sécurité des pipelines CI/CD

Protection de l'intégrité des artefacts (2/2)

CICD-AIP-03 : Vérifications d'intégrité des artefacts et des signatures

Applicable pour : Gestion du code source, registres d'artefacts.

- Mettre en place des mécanismes pour garantir l'intégrité des données, tels que l'utilisation de checksums, de hachages ou de signatures numériques d'artefacts (commits de code source, images conteneurs) pour vérifier l'authenticité des modifications et prévenir les injections de code non autorisées.
- Vérifier régulièrement les données par rapport aux contrôles d'intégrité afin de détecter toute modification ou corruption non autorisée.

CICD-AIP-04 : Chiffrement des artefacts

Applicable pour : Gestion du code source, registres d'artefacts.

- Chiffrer les artefacts confidentiels (à minima) lorsqu'ils sont stockés à l'aide d'un algorithme de chiffrement fort.
- S'assurer que les certificats de chiffrement sont à jour et gérés de manière sécurisée.
- Réviser et mettre régulièrement à jour les protocoles de chiffrement pour les aligner sur les normes de sécurité actuelles.

Surveillance et détection

CICD-MON-01 : Journalisation de sécurité

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Permettre la journalisation des actions critiques effectuées via les composants CI/CD, afin de servir de traces auditées.

CICD-MON-02 : Surveillance et détection d'incidents

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Activer les alertes natives pour surveiller les activités sensibles sur les composants CI/CD.
- Envisager une intégration des composants CI/CD dans des capacités de détection centralisées (ex :SIEM).



Principes directeurs de sécurité des pipelines CI/CD

Continuité et reprise de l'activité

CICD-COR-01 : Gestion des backups

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- S'assurer de la sauvegarde des composants CI/CD et de leur configuration (p. ex. état de provisionnement et fichiers descripteurs d'infrastructure *-playbook/manifest/recipe-*, modules *Infrastructure-as-Code*), ainsi que des artefacts (avec chiffrement approprié et stockage sécurisé).
- S'assurer que les sauvegardes peuvent être restaurées avec succès conformément les objectifs de temps de reprise (RTO) et les objectifs de points de reprise (RPO) définis.

CICD-COR-02 : Plan de reprise après sinistre

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- Mettre en place un plan de reprise après sinistre (*Disaster Recovery Plan*) pour restaurer les composants et artefacts CI/CD en cas d'incident.
- Mettre régulièrement à jour et tester le plan de reprise (p. ex. annuellement ou après des changements importants).

Assurance sécurité

CICD-SAC-01 : Audits réguliers des composants CI/CD

Applicable pour : tout composant CI/CD (en priorité les SCM, orchestrateur CI/CD et registres d'artefacts).

- S'assurer que les composants et modèles CI/CD sont régulièrement audités afin de détecter d'éventuelles mauvaises configurations, vulnérabilités ou droits et permissions abusifs.
- Effectuer un test d'intrusion des composants et modèles CI/CD avant la mise en service et après un changement majeur afin de s'assurer qu'aucune vulnérabilité de sécurité n'a été introduite. Un changement majeur peut être applicatif (p. ex. nouvelle interface, fonctionnalité métier, version majeure, etc.) ou technique (p. ex. nouvelle architecture, infrastructure, changement de protocole, etc.).



Exemple de contrôle d'accès s'appuyant sur les rôles pour la chaîne d'outils CI/CD

Les recommandations RBAC* suivantes pour la CI/CD constituent une **base à décliner, adapter et/ou préciser** localement en fonction de l'organisation, des technologies et de l'appétence au risque.

- **Développeur** : se concentre sur le codage et les contributions quotidiennes, avec un accès limité aux branches protégées et aux modifications de l'infrastructure critique.
- **Développeur expérimenté** : développeur habilité à fusionner des *pull/merge requests* et à pousser vers des branches protégées, prenant en charge les revues de code et l'intégrité du dépôt. Ce rôle devrait être limité à une ou deux personnes par équipe de développement.
- **Ingénieur DevOps** : gère les workflows CI/CD, les pipelines de déploiement et la promotion des artefacts. Dispose d'autorisations étendues sur les aspects opérationnels, sans pour autant disposer de droits d'administration complets.
- **Administrateur** : gère les paramètres du dépôt, les permissions et les journaux d'audit. Dispose d'un contrôle total sur le code source (SCM) et les artefacts. Le nombre de comptes privilégiés associés doit être limité.

Activités		Rôles			
Outil	Description	Développeur	Développeur expérimenté	Ingénieur DevOps	Administrateur
Gestion du code source	Cloner un dépôt de code	✓	✓	✓	✓
	Pousser du code directement vers des branches non protégées	✓	✓	✓	✓
	Pousser du code vers des branches protégées	✗	✓	✗	✓
	Créer et modifier des workflows CI/CD	✗	✗	✓	✓
	Approuver les <i>pull/merge requests</i> vers des branches protégées	✗	✓	✗	✓
	Gérer les paramètres du dépôt de code	✗	✓	✓	✓
	Gérer les règles de protection des branches	✗	✗	✗	✓
	Consulter les journaux de sécurité / journaux d'audit	✓	✓	✓	✓
	Gérer les utilisateurs, les droits d'accès et les paramètres généraux du SCM	✗	✗	✗	✓
Registres d'artefacts	Publier des artefacts	✗	✓	✓	✓
	<i>Pull</i> /téléchargement des artefacts	✓	✓	✓	✓
	Gérer les registres d'artefacts (créer/supprimer)	✗	✗	✓	✓
	Gérer des autorisations pour les registres	✗	✗	✗	✓
	Déployer des pipelines de promotion d'artefacts	✗	✗	✓	✓
	Configurer les politiques de conservation du registre des artefacts	✗	✗	✓	✓
	Consulter les logs de sécurité/logs d'audit	✓	✓	✓	✓
	Gérer les utilisateurs du Registre des artefacts, les droits d'accès et les paramètres généraux	✗	✗	✗	✓

Contrôles de sécurité	Comptes basés sur les rôles CI/CD			
	Description	Développeur	Développeur principal	Ingénieur DevOps
Mettre en place de la politique d'authentification et activer le SSO (<i>Single Sign-On</i>)	✗	✗	✗	✓
Activer l'authentification multifactorielle (MFA)	✓	✓	✓	✓
Centralisation de la gestion des comptes privilégiés	✗	✗	✗ <i>(intéressant à avoir)</i>	✓
Suivre régulièrement des formations de sécurité et des sensibilisations sur les actions sensibles et impacts potentiels	✓	✓	✓	✓
Limiter le nombre de comptes	✗	✓	✓	✓
Revue régulière des comptes, rôles et permissions	✗	✓	✗	✓



Gestionnaire de code source – Risques

Un outil de gestion du code source (SCM) fournit le contrôle de version, l'intégration continue et des outils de collaboration au sein d'un environnement intégré unique. Il rationalise les workflows de développement, garantissant une gestion efficace du code, une collaboration sécurisée et des pipelines CI/CD robustes.

Exemples d'outils :

Open source : [GitLab CE](#), [Gitea](#), [Gogs](#), [Forgeio](#), [Codeberg](#)...

Commerciaux : [GitHub](#), [GitLab EE](#), [Bitbucket](#), [Tuleap](#), [Tuleap Azure DevOps Repos](#), [AWS CodeCommit](#), [Google Cloud Source repositories](#)...

ID	Description du risque	Impact	Probabilité	Référence Mitre Att&ck	Vulnérabilités liées au risque (OWASP Top 10 CI/CD ; CWE & CVE)
R01	Exfiltration d'informations sensibles à partir des dépôts SCM et/ou des logs CI/CD en raison d'une protection des données ou d'une configuration inadéquate des dépôts (p. ex. dépôts publics)			T1074 (Data Staged)	OWASP: Insecure System Configuration CWE-200 (Information Exposure) CVE-2023-2620 , CVSS: 5.5
R02	Modification ou suppression non autorisée d'artefacts ou de code source au sein des dépôts et registres GitLab, due à des contrôles d'accès insuffisants ou à des permissions mal configurées.			T1070 (Indicator Removal)	OWASP: Insufficient PBAC (Pipeline-Based Access Controls) CWE-285 (Improper Authorization) CVE-2023-2576 , CVSS: 4.3
R03	Attaque par déni de service (DoS) sur les instances SCM entraînant une perturbation du service			T1499 (Endpoint Denial of Service)	OWASP: Insufficient Flow Control Mechanisms CWE-770 (Allocation of Resources Without Limits or Throttling) CVE-2024-9631 , CVSS: 7,5
R04	Actions non intentionnelles dues à des déclencheurs de tâches CI/CD mal configurés ou des règles d'automatisation (perte de données, dérive de configuration, état incohérent...)			T1059 (Command and Scripting Interpreter)	OWASP: Insecure System Configuration; Ungoverned Usage of CI/CD Pipelines CWE-427 (Uncontrolled Search Path Element) CVE-2022-24765 , CVSS: 6.0
R05	Exploitation des vulnérabilités dans la SCM et ses composants tiers (exploitation de faiblesses telles que l'injection SQL, le scripting cross-site, l'exécution de code à distance...)			T1190 (Exploit Public-Facing Application)	OWASP: Poisoned Pipeline Execution (PPE) CWE-22 (Path Traversal) CVE-2023-2825 , CVSS: 10.0



Gestionnaire de code source – Recommandations (1/2)

Exemples d'outils :

Open source : [GitLab CE](#), [Gitea](#),
[Gogs](#), [Forgejo](#), [Codeberg](#)...

Commerciaux : [GitHub](#), [GitLab EE](#), [Bitbucket](#), [Tuleap](#), [Tuleap Azure DevOps Repos](#), [AWS CodeCommit](#), [Google Cloud Source repositories](#)...

Un outil de gestion du code source (SCM) fournit le contrôle de version, l'intégration continue et des outils de collaboration au sein d'un environnement intégré unique. Il rationalise les workflows de développement, garantissant une gestion efficace du code, une collaboration sécurisée et des pipelines CI/CD robustes.

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLSA**
REC01	R01	Protection des données	Mettre en œuvre des mesures de sécurité robustes pour contrôler l'accès au SCM et protéger les données sensibles au repos et en transit.	++	Level 2 Security testing	Level 2 deployment
REC02	R05	Patching	Mettre à jour régulièrement le SCM et les composants tiers afin de corriger les vulnérabilités connues.	+	Level 2 Environment management	Level 2 build
REC03	R01, R02	IAM	Configurer des contrôles d'accès stricts pour protéger les configurations et données sensibles du SCM.	++	Level 2 Security architecture	Level 3 source control
REC04	R03	Réseau	Mettre en œuvre des mesures de sécurité réseau telles que la limitation de débit et des services de protection contre les attaques DDoS pour sécuriser les serveurs SCM.	+++	Level 2 Operational management	Level 3 deployment
REC05	R01, R02	IAM	Configurer correctement les rôles et permissions (RBAC) afin de limiter les accès selon le principe du moindre privilège, et revoir régulièrement les droits d'accès des utilisateurs ainsi que les fonctionnalités de collaboration sécurisée.	++	Level 2 Security architecture	Level 3 source control
REC06	R04	Configuration	Éviter les configurations par défaut et réaliser des audits de sécurité et des analyses de configuration réguliers des paramètres du SCM.	++	Level 2 Architecture assessment	Level 3 source control
REC07	R04, R02	Configuration	Mettre en œuvre des contrôles d'accès stricts et recourir au contrôle de version pour les configurations et paramètres du SCM.	++	Level 3 Secure build	Level 3 source control



Gestionnaire de code source – Recommandations (2/2)

Un outil de gestion du code source (SCM) fournit le contrôle de version, l'intégration continue et des outils de collaboration au sein d'un environnement intégré unique. Il rationalise les workflows de développement, garantissant une gestion efficace du code, une collaboration sécurisée et des pipelines CI/CD robustes.

Exemples d'outils :

Open source : [GitLab CE](#), [Gitea](#), [Gogs](#), [Forgejo](#), [Codeberg](#)...

Commerciaux : [GitHub](#), [GitLab EE](#), [Bitbucket](#), [Tuleap](#), [Tuleap Azure DevOps Repos](#), [AWS CodeCommit](#), [Google Cloud Source repositories](#)...

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLSA**
REC08	R04	Pipelines	Mettre régulièrement à jour et revoir les pipelines et scripts du Source Code Manager afin de s'assurer qu'ils respectent les meilleures pratiques de sécurité	++	Level 3 Security testing	Level 3 build
REC09	R03, R04	Sensibilisation	Renforcer la formation à la sécurité et la réponse aux incidents pour les utilisateurs et administrateurs du gestionnaire de code source	++	Level 2 Governance	Level 2 source control



Orchestrateur d'intégration continue – Risques

Exemples d'outils :

Open source : [Jenkins](#), [GitLab CE](#), [Spinnaker](#)...

Commerciaux : [Github actions](#), [Bitbucket Pipelines](#), [Travis CI](#), [Azure Pipelines](#), [AWS CodePipeline](#), [Google Cloud Build](#)...

L'orchestration de l'intégration continue (CIO) automatise la coordination des pipelines CI à travers plusieurs étapes, outils et environnements.

ID	Description du risque	Impact	Probabilité	Référence Mitre Att&ck	Vulnérabilités liées au risque (OWASP Top 10 CI/CD ; CWE & CVE)
R01	Empoisonnement des pipelines CIO via une injection de commandes dans les scripts de tâches	Yellow	Red	T1059 (Command and Scripting Interpreter)	OWASP: Poisoned Pipeline Execution (PPE) CWE-94 (Code Injection) CVE-2018-1000861 , CVSS: 9.8
R02	Exfiltration de données sensibles à partir des logs de compilation et dépôts CIO en raison de mécanismes de protection inadéquats	Yellow	Green	T1074 (Data Staged)	OWASP: Insufficient Credential Hygiene CWE-200 (Information Exposure) CVE-2020-2103 , CVSS: 5.4
R03	Vulnérabilités de sécurité liées à l'utilisation de plugins obsolètes ou vulnérables	Yellow	Yellow	T1505 (Server Software Component)	OWASP: Dependency Chain Abuse CWE-829 (Inclusion of Functionality from Untrusted Control Sphere) CVE-2022-30954 , CVSS: 6.5
R04	Modification ou suppression non autorisée d'artefacts de compilation en raison de contrôles d'accès insuffisants	Red	Yellow	T1070 (Indicator Removal)	OWASP: Insufficient PBAC (Pipeline-Based Access Controls) CWE-284 (Improper Access Control) CVE-2018-1999003 , CVSS: 4.3
R05	Risques de sécurité liés à une gestion inadéquate des correctifs pour les serveurs et plugins	Red	Red	T1601 (Exploitation for Client Execution)	OWASP: Insecure Update Mechanism CWE-732 (Incorrect Permission Assignment for Critical Resource) CVE-2024-23897 , CVSS: 9.8



Orchestrateur d'intégration continue – Recommandations

Exemples d'outils :

Open source : [Jenkins](#), [GitLab CE](#), [Spinnaker](#)...

Commerciaux : [Github actions](#), [Bitbucket Pipelines](#), [Travis CI](#), [Azure Pipelines](#), [AWS CodePipeline](#), [Google Cloud Build](#)...

L'orchestration de l'intégration continue (CIO) automatise la coordination des pipelines CI à travers plusieurs étapes, outils et environnements.

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLSA**
REC01	R03, R05	Patching	Maintenir l'ensemble des plugins à jour et les examiner afin d'identifier les vulnérabilités connues.	+	Level 1 Environment management	Level 2 build
REC02	R02	Protection des données	Stocker et gérer les données sensibles, telles que les identifiants et les secrets, de manière sécurisée au sein de Jenkins.	++	Level 2 Secure build	Level 3 build
REC03	R04	IAM	Définir des permissions granulaires pour les utilisateurs et les jobs afin de limiter les accès selon le principe du moindre privilège (RBAC).	++	Level 2 Security architecture	Level 2 source control
REC04	R01	Logging	Mettre en œuvre des mécanismes d'audit pour les jobs et les scripts afin de suivre les modifications, les exécutions et les interactions des utilisateurs.	++	Level 3 Architecture assessment	Level 2 deployment
REC05	R01	Pipelines	Configurer les jobs pour qu'ils s'exécutent dans des environnements isolés avec des mesures de sécurité appropriées.	+++	Level 3 Security testing	Level 3 build
REC06	R01	Pipelines	Utiliser le Script Security Plugin avec des paramètres correctement configurés pour prévenir l'exécution de scripts non autorisés.	++	Level 3 Secure build	Level 3 build
REC07	R03, R05	Pipelines	Mettre à jour et revoir régulièrement les pipelines, plugins et scripts de l'orchestrateur CI afin de s'assurer qu'ils sont à jour et sécurisés.	++	Level 2 Security testing	Level 3 build



Gestion des artefacts – Risques

La gestion des artefacts assure le stockage, la gestion des versions et la distribution des sorties de *build* (binaires, images Docker, packages) produites lors de l'intégration continue.

Exemples d'outils :

Open source : [JFrog Artifactory OSS](#), [Harbor](#), [GitLab Package Registry \(CE\)](#)...

Commerciaux : [JFrog Artifactory](#), [Sonatype Nexus Repository](#), [GitHub Packages](#), [Azure Artifacts](#), [AWS CodeArtifact](#), [Google Artifact Registry](#)...

ID	Description du risque	Impact	Probabilité	Référence Mitre Att&ck	Vulnérabilités liées au risque (OWASP Top 10 CI/CD ; CWE & CVE)
R01	Exfiltration de données sensibles via des registres et logs mal protégés (mécanismes de contrôle d'accès insuffisants, absence de mécanismes de détection de données sensibles/secrètes)			T1074 (Data Staged)	OWASP: Insufficient Credential Hygiene CWE-200 (Information Exposure) CVE-2023-42661 , CVSS: 7.2
R02	Utilisation de plugins vulnérables , menant à des attaques via ces extensions			T1505 (Server Software Component)	OWASP: Dependency Chain Abuse CWE-829 (Inclusion of Functionality from Untrusted Control Sphere) CVE-2024-2247 , CVSS: 8.8
R03	Utilisation de configurations par défaut ou incorrectes (paramètres du dépôt, jetons d'accès, politiques de gestion des artefacts...), entraînant des vulnérabilités de sécurité			T1078 (Valid Accounts)	OWASP: Security Misconfiguration CWE-276 (Incorrect Default Permissions) CVE-2019-17444 , CVSS: 9.8
R04	Surveillance et logging insuffisants , entraînant des activités suspectes non détectées ou des incidents de sécurité			T1105 (Ingress Tool Transfer)	OWASP: Insufficient Logging and Visibility CWE-778 (Insufficient Logging) CVE-2023-42661 , CVSS: 7.2
R05	Gestion inadéquate des correctifs pour le serveur de gestion d'artefact et des plugins , entraînant l'exploitation de vulnérabilités connues			T1070 (Indicator Removal)	OWASP: Vulnerable and Outdated Components CWE-200 (Exposure of Sensitive Information to an Unauthorised Actor) CVE-2022-0573 , CVSS: 8.8



Gestion des artefacts – Recommandations

La gestion des artefacts assure le stockage, la gestion des versions et la distribution des sorties de *build* (binaires, images Docker, packages) produites lors de l'intégration continue.

Exemples d'outils :

Open source : [GitLab Package Registry \(CE\)](#), [docker/container registry](#), [JFrog Artifactory OSS](#), [Harbor](#), ...

Commerciaux : [GitHub Packages](#), [JFrog Artifactory](#), [Sonatype Nexus Repository](#), [Azure Artifacts](#), [AWS CodeArtifact](#), [Google Artifact Registry](#)...

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLSA**
REC01	R03	Configuration	Effectuer des revues de sécurité et des audits réguliers des configurations du gestionnaire d'artefacts, incluant le scan de vulnérabilités des artefacts.	++	Level 2 Architecture assessment	Level 3 build
REC02	R01	Protection des données	Utiliser le chiffrement pour protéger les données sensibles, notamment les identifiants d'accès et les paramètres de configuration.	++	Level 2 Secure build	Level 3 deployment
REC03	R01	Protection des données	S'assurer que les artefacts sont chiffrés et signés afin de protéger leur intégrité et leur confidentialité tout au long de leur cycle de vie.	+++	Level 3 Security testing	Level 3 deployment
REC04	R04	Logging	Activer une journalisation exhaustive et revoir régulièrement les journaux relatifs aux activités des utilisateurs, aux événements système et aux détails d'accès.	++	Level 2 Incident management	Level 3 deployment
REC05	R04	Logging	Utiliser des mécanismes de détection d'anomalies dans les journaux afin d'identifier et de traiter les activités inhabituelles ou potentiellement malveillantes.	+++	Level 3 Incident management	Level 3 deployment
REC06	R05, R02	Patching	Mettre à jour régulièrement l'application et ses plugins afin de corriger les vulnérabilités connues et maintenir la sécurité du système.	++	Level 1 Environment management	Level 2 build
REC07	R04	Configuration	Utiliser des systèmes de contrôle de version pour gérer et suivre les modifications apportées aux configurations, dans un souci de traçabilité et de cohérence.	++	Level 2 Security architecture	Level 2 source control



Provisionnement – Risques

Le provisionnement automatise la création et la configuration des ressources d'infrastructure (machines virtuelles, conteneurs, réseaux, bases de données) nécessaires aux pipelines CI/CD et au déploiement des applications.

Exemples d'outils :

Open source : [Ansible](#), [Open Source Puppet](#), [OpenVox](#), [OpenTofu](#), [Chef](#), [pyinfra](#)...

Commerciaux : [Terraform Enterprise](#), [perforce Puppet](#), [AWS CloudFormation](#), [Azure Resource Manager](#)...

ID	Description du risque	Impact	Probabilité	Référence Mitre Att&ck	Vulnérabilités liées au risque (OWASP Top 10 CI/CD ; CWE & CVE)
R01	Accès non autorisé aux fichiers descripteurs d'infrastructure (<i>playbook/manifest/recipe</i>) et aux fichiers d'inventaire en raison d'autorisations utilisateur mal configurées	Élevé	Moyenne	T1078 (Valid Accounts)	OWASP: Inadequate Authentication & Access Management; Security Misconfiguration CWE-287 (Improper Authentication) CVE-2020-1746 , CVSS: 5.0
R02	Empoisonnement des fichiers descripteurs d'infrastructure (<i>playbook/manifest/recipe</i>) via une injection de commandes, permettant d'exécuter des commandes malveillantes	Élevé	Moyenne	T1059 (Command and Scripting Interpreter)	OWASP: Poisoned Pipeline Execution (PPE); Insufficient Flow Control Mechanisms CWE-94 (Code Injection) CVE-2020-14332 , CVSS: 5.5
R03	Exfiltration de données sensibles via des fichiers descripteurs d'infrastructure (<i>playbook/manifest/recipe</i>), fichiers d'inventaire ou logs mal protégés (mécanismes de contrôle d'accès insuffisants, manque de mécanismes de détection de données sensibles/secrètes)	Élevé	Moyenne	T1074 (Data Staged)	OWASP: Insecure Data Management; Insufficient Credential Hygiene CWE-200 (Information Exposure) CVE-2019-10217 , CVSS: 5.7
R04	Utilisation de rôles ou modules vulnérables, menant à des attaques via ces composants	Élevé	Moyenne	T1505 (Server Software Component)	OWASP: Dependency Chain Abuse; Ungoverned Usage of 3rd Party Services CWE-829 (Inclusion of Functionality from Untrusted Control Sphere) CVE-2021-3620 , CVSS: 5.5
R05	Modification ou suppression non autorisée de fichiers descripteurs d'infrastructure (<i>playbook/manifest/recipe</i>) ou de configurations en raison de contrôles d'accès insuffisants	Élevé	Moyenne	T1070 (Indicator Removal)	OWASP: Insufficient PBAC (Pipeline-Based Access Controls) CWE-284 (Improper Access Control) CVE-2019-10180 , CVSS: 4.8



Approvisionnement – Recommandations

Le provisionnement automatise la création et la configuration des ressources d'infrastructure (machines virtuelles, conteneurs, réseaux, bases de données) nécessaires aux pipelines CI/CD et au déploiement des applications.

Exemples d'outils :

Open source : [Ansible](#), [Open Source Puppet](#), [OpenVox](#), [OpenTofu](#), [Chef](#), [pynfra](#)...

Commerciaux : [Terraform Enterprise](#), [perforce Puppet](#), [AWS CloudFormation](#), [Azure Resource Manager](#)...

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLSA**
REC01	R01	IAM	Mettre en place des mécanismes d'authentification et d'autorisation appropriés pour restreindre l'accès à la solution de provisionnement et à ses ressources.	++	Level 2 Security architecture	Level 2 source control
REC02	R03	Protection des données	Utiliser le chiffrement pour les données sensibles contenues dans les fichiers descripteurs d'infrastructure (<i>playbook/manifest/recipe</i>), fichiers d'inventaire et journaux, afin de prévenir tout accès non autorisé.	++	Level 2 Secure build	Level 3 deployment
REC03	R04	Patching	Maintenir l'ensemble des rôles et modules de la solution de provisionnement à jour et les examiner afin d'identifier les vulnérabilités.	++	Level 1 Environment management	Level 2 build
REC04	R04	Patching	Mettre à jour régulièrement l'outil et ses composants afin de corriger les vulnérabilités connues et maintenir la sécurité du système.	+	Level 1 Environment management	Level 2 build
REC05	R01	Configuration	Auditer régulièrement les configurations et permissions de la solution de provisionnement afin de prévenir les mauvaises configurations et garantir des contrôles d'accès appropriés.	++	Level 2 Architecture assessment	Level 3 source control
REC06	R02	Pipelines	Appliquer des pratiques de codage sécurisé pour les fichiers descripteurs d'infrastructure (<i>playbook/manifest/recipe</i>) et les rôles afin de prévenir les injections de commandes et autres problèmes de sécurité.	++	Level 2 Secure build	Level 3 build
REC07	R05	Configuration	Utiliser des systèmes de contrôle de version pour gérer et suivre les modifications apportées aux configurations de la solution de provisionnement, dans un souci de traçabilité et de cohérence.	++	Level 2 Security architecture	Level 2 source control
REC08	R04	Configuration	S'assurer que le code obsolète ou non maîtrisé (rôles, modules, fichiers descripteurs d'infrastructure) fait l'objet d'une revue régulière et est supprimé si nécessaire.	++	Level 2 Requirements driven testing	Level 3 build



Conteneurisation et Ordonnancement – Risques

La conteneurisation et l'ordonnancement gèrent les workloads et les services conteneurisés, en optimisant les opérations de déploiement et d'exécution.

Exemples d'outils :

Open source : [docker](#), [podman](#), [AppTainer](#), [Linux Container \(Incus, LXC...\)](#), [Kubernetes](#), [docker swarm](#)...

Commerciaux : [RedHat OpenShift](#), [Amazon Elastic Kubernetes Service](#), [Azure Kubernetes Service](#), [Google Kubernetes Engine](#), [Suse Rancher](#)...

ID	Description du risque	Impact	Probabilité	Référence Mitre Att&ck	Vulnérabilités liées au risque (OWASP Top 10 CI/CD ; CWE & CVE)
R01	Empoisonnement du pipeline via la falsification des manifestes de configuration de l'orchestration des conteneurs permettant l'exécution de commandes malveillantes dans les conteneurs éphémères (<i>Pods</i>) déployés	Élevé	Moyenne	T1059 (Command and Scripting Interpreter)	OWASP: Poisoned Pipeline Execution (PPE); Insufficient Flow Control Mechanisms CWE-94 (Code Injection) CVE-2019-11253 , CVSS: 7,5
R02	Exfiltration de données sensibles via leurs exposition au sein de configMaps , lesquels n'ont pas été conçus pour stocker des données confidentielles	Élevé	Moyenne	T1074 (Data Staged)	OWASP: Insecure Data Management; Insufficient Credential Hygiene CWE-200 (Information Exposure) CVE-2024-3177 , CVSS: 2.7
R03	Utilisation d'images de conteneurs vulnérables , menant à des attaques via ces images	Élevé	Élevé	T1505 (Server Software Component)	OWASP: Dependency Chain Abuse; Ungoverned Usage of 3rd Party Services CWE-829 (Inclusion of Functionality from Untrusted Control Sphere) CVE-2020-8554 , CVSS: 6,6
R04	Modification ou suppression non autorisée de ressources (<i>Pods</i> , rôles de cluster...) en raison de contrôles d'accès insuffisants	Élevé	Moyenne	T1070 (Indicator Removal)	OWASP: Insufficient PBAC (Pipeline-Based Access Controls) CWE-284 (Improper Access Control) CVE-2021-25741 , CVSS: 8.8
R05	Isolation insuffisante des <i>Pods</i> menant à des attaques potentielles entre conteneurs puis à une potentielle compromission de l'hôte (par échappement de conteneurs)	Élevé	Moyenne	T1071: Application Layer Protocol	OWASP: Security Misconfiguration CWE-668: Exposure of Resource to Wrong Sphere CVE-2023-5528 , CVSS: 7.2



Conteneurisation et Ordonnancement – Recommandations

Exemples d'outils :

Open source : [docker](#), [podman](#), [AppTainer](#), [Linux Container \(Incus, LXC...\)](#), [Kubernetes](#), [docker swarm...](#)

Commerciaux : [RedHat OpenShift](#), [Amazon Elastic Kubernetes Service](#), [Azure Kubernetes Service](#), [Google Kubernetes Engine](#), [Suse Rancher...](#)

La conteneurisation et l'ordonnancement gèrent les workloads et les services conteneurisés, en optimisant les opérations de déploiement et d'exécution.

ID	ID du risque	Thème	Recommandations	Complexité	Niveau SAMM*	Niveau SLISA**
REC01	R05	Réseau	Mettre en œuvre des politiques réseau pour contrôler et restreindre le trafic entre les pods, renforçant ainsi la sécurité et réduisant la surface d'attaque.	+++	Level 2 Operational management	Level 3 deployment
REC02	R02	Protection des données	Stocker des données sensibles et confidentielles au sein de coffres-forts de secrets (secret , vault , hiera , etc.) et ne jamais stocker de données confidentielles dans les ConfigMaps , afin de protéger les données contre tout accès non autorisé.	++	Level 2 Secure build	Level 3 deployment
REC03	R04	IAM	Utiliser le contrôle d'accès basé sur les rôles (RBAC) et d'autres mécanismes d'autorisation pour gérer l'accès aux ressources et actions de la solution de conteneurisation et d'ordonnancement.	+++	Level 2 Security architecture	Level 3 source control
REC04	R05	Pipelines	Définir et appliquer des contextes de sécurité pour les pods de la solution de conteneurisation et d'ordonnancement, afin de garantir une isolation appropriée, limiter les privilèges des conteneurs et respecter le principe du moindre privilège.	+++	Level 2 Security architecture	Level 3 build
REC05	R03	Patching	Analyser les images de conteneurs à la recherche de vulnérabilités et s'assurer qu'elles sont signées afin de vérifier leur intégrité et leur sécurité.	++	Level 2 Security testing	Level 3 build
REC06	R01	Configuration	Passer en revue et sécuriser (p. ex. signature de <i>commits</i> et vérifications de ces signatures, application systématique d'outils d'analyse -linter, scanner, etc.-) les manifestes de la solution de conteneurisation et d'ordonnancement afin de prévenir les injections de commandes et les modifications non autorisées.	++	Level 2 Requirements driven-testing	Level 3 build
REC07	R04	Journalisation	Utiliser des outils et des pratiques permettant de détecter les anomalies et les comportements inhabituels dans les opérations et les journaux de la solution de conteneurisation et d'ordonnancement.	+++	Level 3 Incident management	Level 3 deployment



PIPELINES CI/CD – FEUILLES DE ROUTE DE HAUT NIVEAU



Trois profils d'entreprises

Cette étude propose des feuilles de route de sécurité CI/CD graduelles pour trois profils d'entreprise.



1 – Start up

Équipes de taille réduite, évoluant rapidement, généralement cloud-native et s'appuyant sur des plateformes CI/CD SaaS modernes.

La sécurité est souvent reléguée au second plan au profit de la rapidité et de la livraison produit, avec peu de processus formalisés. L'enjeu principal est d'intégrer des garde-fous de sécurité simples mais robustes dès les premières étapes, afin que la croissance n'amplifie pas les pratiques non sécurisées.



2 – Petites et moyennes entreprises

Entreprises de taille intermédiaire avec une activité de développement croissante et des effectifs dédiés à la sécurité limités.

Elles s'appuient généralement sur un mix de services cloud managés et de quelques outils CI/CD internes. Leur priorité est d'industrialiser rapidement et efficacement la sécurité CI/CD, en recourant à une automatisation pragmatique et à des pratiques standardisées.



3 – Grande entreprise

Grandes organisations fortement régulées, disposant d'infrastructures hybrides complexes et de multiples équipes de développement.

Elles exploitent souvent plusieurs plateformes CI/CD en parallèle, combinant des stacks on-premise et cloud-native. L'enjeu est de sécuriser le CI/CD à grande échelle, avec une traçabilité forte, une intégration avec les outils de sécurité existants et des exigences de conformité strictes.



Start-ups

Les start-ups démarrent avec des moyens limités mais doivent adopter une approche shift-left dès le départ (en s'appuyant sur des outils gratuits et open source, et en intégrant la sécurité dès le premier jour), pour faire évoluer leur outillage au fur et à mesure de leur croissance.

Quick wins

Sécurité dès le premier jour

- Interdire l'inclusion de secrets dans le code en utilisant les gestionnaires de secrets GitHub/GitLab ou un coffre-fort.
- Exécuter des scanners de secrets en pré-commit (par ex. git-secrets, TruffleHog) sur tous les dépôts.

Hygiène des dépôts

- Définir un modèle de branches clair et protéger les déploiements en production.
- Exiger une revue par les pairs avant de merger dans la branche principale.
- Mettre en place des branches protégées et interdire les pushes directs vers main/master.

Durcissement des outils

- Mettre en place le MFA, un RBAC strict à la fois sur outils et dépôts, et maintenir des sauvegardes de code externes.

S'appuyer sur les outils gratuits

- Utiliser des services open source et gratuits (p. ex. GitHub CodeQL ou SonarQube Community pour le SAST et npm audit/security pour les dépendances).
- Utiliser les alertes de vulnérabilité GitHub/GitLab.
- Générer des SBOMs pour les builds

Journalisation de base

- Activer les logs d'audit et les notifications dans la CI/CD (la plupart des outils SaaS CI proposent des logs basiques).
- Utiliser un simple collecteur de logs (alertes Slack/email ou ELK gratuit) pour les erreurs de compilation ou les activités suspectes.

Moyen terme

Gestion des secrets

- Automatiser la distribution des secrets dans le pipeline CI (p. ex. Vault-env, SecretHub, injection de secrets cloud).
- Migrer vers une plateforme de gestion des secrets de base (p. ex. intégration Vault CLI ou KMS cloud) afin que les développeurs ne manipulent plus directement les clés en clair.

Identifiants sur demande

- Adopter les intégrations OIDC*/IdP** (p. ex. les *GitHub Actions* assument des rôles AWS sans stocker les clés AWS).

Contrôles de sécurité automatisés

- Réaliser le scan d'images des conteneurs (p. ex. *Trivy*, *Skopeo*, *Grype*, *Syft*, *Clair*) avant déploiement.
- Automatiser les alertes sur toute découverte critique.

Journalisation et alertes

- Commencer à utiliser un SIEM léger ou un outil de log (p. ex. Sumo Logic Free, Datadog Free) pour collecter les logs CI/CD.
- Créer des alertes simples (p. ex. lors de plusieurs tentatives de déploiement) et examiner régulièrement les logs.

Long terme

Culture DevSecOps

- Exiger que tous les artefacts soient signés.
- Utiliser le policy-as-code (p. ex. le pipeline échoue si de nouvelles dépendances ont des CVE connus).

Sécurité augmentée par l'IA

- Exploiter des outils avancés (p. ex. scanners de code assistés par IA, bots de sécurité SaaS) pour assister les efforts d'une petite équipe.

* *OpenID Connect*
** *Identity provider*



Petites et moyennes entreprises

Les PME privilégient généralement les services managés et cloud, ainsi que l'automatisation pour l'efficacité économique.

Quick wins

Sécurité des plateformes

- Utiliser des plateformes CI/CD avec des fonctionnalités de sécurité intégrées.
- Activer la 2FA, la protection des branches et le chiffrement au repos pour les dépôts.
- Exploiter les stockages de secrets intégrés aux plateformes (p. ex. GitHub Secrets, Azure Key Vault pour les pipelines Azure).

Starter Pack DevSecOps

- Activer les modules complémentaires de sécurité gratuits : p. ex. GitHub Code Scanning (CodeQL) ou SonarCloud pour le SAST, Dependabot ou Snyk (version gratuite) pour les vérifications de dépendances.
- Exécuter des tests simples en CI (p. ex. audit npm, bandit, OWASP Dependency-Check) à chaque commit.

Hygiène des identifiants

- Auditer et renouveler tous les identifiants CI par défaut ou partagés.
- Utiliser des rôles IAM Cloud avec le principe du moindre privilège plutôt que des clés codées en dur.
- Supprimer les comptes inutilisés.

Journalisation de base

- S'assurer que les serveurs CI enregistrent les événements clés. Agréger les logs et définir des alertes pour les échecs de compilation ou déploiements en production.

Moyen terme

IAM intégré

- Connecter la CI/CD à l'annuaire d'entreprise (OAuth/OIDC avec Azure/AWS IAM).
- Appliquer un RBAC granulaire dans les outils CI pour que les développeurs n'accèdent qu'à leurs projets.
- Mettre en place des workflows automatisés pour le provisionnement et le déprovisionnement des utilisateurs CI (via l'IdP).

Automatisation des secrets

- Implémenter une solution simple de gestion des secrets (p. ex. [HashiCorp Vault](#), cloud KMS) et automatiser les rotations (p. ex. via Jenkins Credential Store).
- Supprimer les mises à jour manuelles des secrets du processus de release.

Numérisation complète du pipeline

- Intégrer des gates de sécurité automatisées : contrôles IaC (p. ex. [Checkov](#) pour scan Terraform ou Kubernetes), scans d'images de conteneurs (p. ex. [Clair/Trivy](#)). Bloquer les *builds* en cas de vulnérabilités critiques.

Surveillance à grande échelle

- Renforcer la télémétrie : corréler les logs CI/CD avec les logs applicatifs.
- Utiliser la journalisation cloud-native (par ex. CloudWatch, Azure Monitor) ou un SIEM pour détecter les anomalies.
- Mettre en place des fichiers descripteurs d'infrastructure (playbook/manifest/recipe) de base (p. ex. notification automatique en cas de multiples échecs de pipeline).

Long terme

Assurance des pipelines

- Exiger la signature des commits ou le tagging de chaque release et imposer la génération systématique de SBOM.
- Adopter la conformité-as-code : les pipelines appliquent automatiquement les politiques de sécurité (p. ex. via Open Policy Agent/Gatekeeper).

Amélioration des outils

- Intégrer des outils plus avancés en fonction du budget disponible (p. ex. SAST/SCA sous licence tels que Snyk Enterprise, Twistlock/Aqua pour les conteneurs). Exploiter leurs intégrations CI pour bloquer les modifications à risque.

Surveillance continue

- Étendre la sécurité au-delà du *build* : intégrer la supervision runtime (p. ex. [Falco](#) ou [Cilium](#) pour la supervision noyau s'appuyant sur eBPF pour les conteneurs, WAF pour le runtime) et remonter les vulnérabilités vers la CI (p. ex. bloquer le *build* si une vulnérabilité est détectée après déploiement).

Exploiter les mécanismes de sécurité natifs du fournisseur cloud

- Éviter les VM génériques avec des accès larges : associer un rôle IAM spécifique à un compte de service Kubernetes pour que chaque *pod* hérite uniquement des permissions nécessaires.
- Réduire la surface d'attaque en migrant les pipelines CI anciens vers des jobs exécutés dans des *Pods* Kubernetes éphémères ou en utilisant des services *serverless*.



Grandes entreprises

Les grandes entreprises peuvent investir dans des solutions robustes (p. ex. clusters Vault entreprise, SIEM complet, multiples outils de sécurité) et doivent prioriser l'intégration avec les systèmes d'identité et d'infrastructure existants.

Quick wins

Gestion des secrets

- Adopter des coffres-forts centralisés (p. ex. HashiCorp Vault, AWS Secrets Manager, Azure Key Vault, IPA vault, Google Cloud Secret Manager, ou des outils qui consomment des secrets de plusieurs clouds comme **Teller** ou **ESO** -*External Secret Operator*-).
- Éliminer tous les identifiants codés en dur.
- Activer le scanning automatisé des secrets (p. ex. GitSecrets, GitLeaks) pour détecter les fuites.

Durcissement du gestionnaire de code source

- Mettre en place des branches protégées, des revues de code obligatoires via PR, la signature des commits et le MFA sur les dépôts.
- Désactiver les auto-merges et restreindre les privilèges de fork et de visibilité.

Analyse du pipeline

- Intégrer des outils d'analyse statique (SAST) et de scan de dépendances (SCA) dans les pipelines CI (p. ex. SonarQube, Snyk/Dependabot).

Journalisation et surveillance

- Commencer à centraliser les logs CI/CD (par ex. envoyer les logs Jenkins/GitLab CI vers ELK/Splunk) pour une meilleure visibilité.
- Mettre en place des tableaux de bord et des alertes pour les *builds* échoués ou les activités de pipeline anormales.

Moyen terme

RBAC & moindre privilège

- Appliquer un contrôle d'accès strict basé sur les rôles dans les outils CI et les comptes Cloud.
- Attribuer à chaque pipeline et job son propre compte de service avec les permissions strictement nécessaires.
- S'assurer que les *runners* de pipeline s'exécutent sans droits root.
- Conduire des revues d'accès régulières et révoquer automatiquement les rôles inutilisés.
- Migrer vers des identifiants éphémères et un accès *just-in-time* : p. ex. tokens OIDC GitHub/Azure DevOps, assignation de rôles Cloud à durée limitée.

Cycle de vie des identités

- Intégrer les systèmes CI avec l'IdP d'entreprise (par ex. Entra ID, Okta) pour gérer les comptes du pipeline de manière centralisée.
- Désactiver les comptes CI partagés ou locaux, appliquer des identités de service uniques et assurer leur désactivation.
- Maintenir un inventaire de toutes les identités du pipeline.

Analyse avancée

- Scans étendus : IaC (**Checkov**, Terraform Validator), conteneurs (**Trivy**), scanning de code continu avec une couverture large
- Gates de sécurité automatisées (blocage en cas de vulnérabilité critique)

Surveillance renforcée

- Corréler les logs de pipeline et les événements de runtime dans un SIEM.
- Mettre en place la détection d'anomalies sur les flux CI/CD.
- Intégrer les alertes CI/CD dans les workflows SecOps.

Long terme

CI/CD Zero-Trust

- Éliminer les clés à longue durée de vie et appliquer le principe *deny-by-default* sur tous les accès aux pipelines.
- Supprimer en continu les identités obsolètes.

Intégrité de la *supply chain*

- S'appuyer sur des cadres reconnus de sécurité de la *supply chain* logicielle (par ex. SLSA, NIST SSDF, OWASP) pour structurer le durcissement de la CI/CD et mettre en œuvre des contrôles avancés tels que la signature des artefacts (par ex. Sigstore ou in-toto) et la génération systématique de SBOM pour tous les *builds*.
- Vérifier l'intégrité de toutes les dépendances via des checksums cryptographiques ou des signatures.

Détection pilotée par l'IA

- Exploiter la *machine learning* pour détecter les anomalies dans les pipelines CI/CD (p. ex. échecs de tests anormaux ou modifications de code inhabituelles) et prioriser les alertes.
- Intégrer des flux de renseignement sur les menaces dans la supervision CI.



FRAMEWORKS SAMM ET SLSA

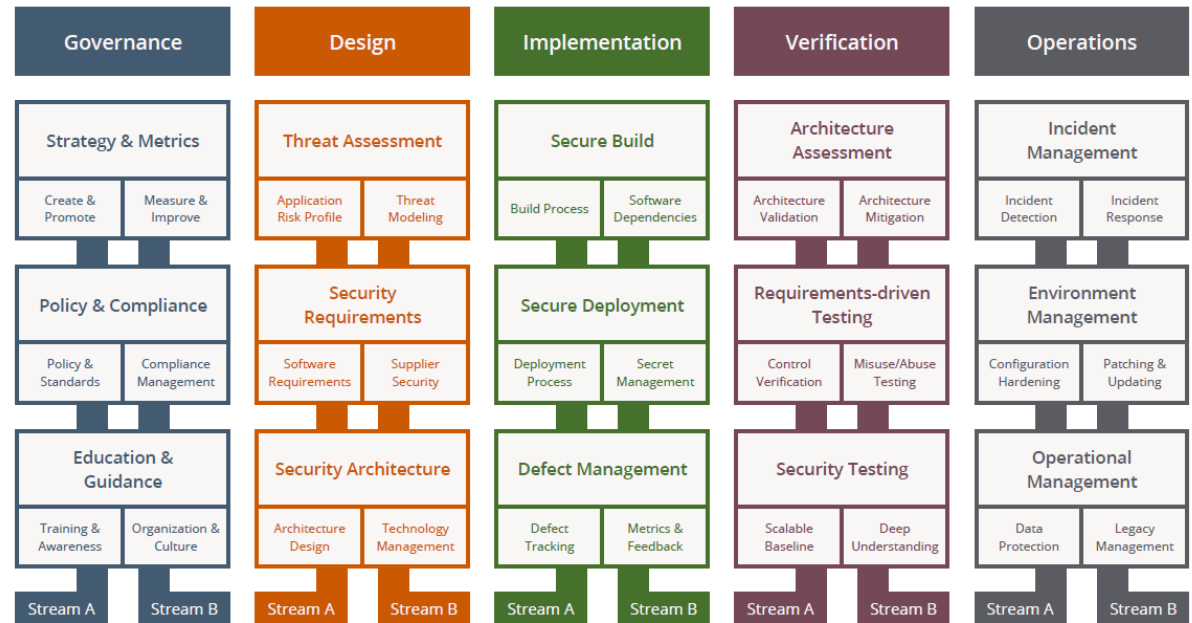


Framework OWASP* Software Assurance Maturity Model (SAMM)

Le *framework* OWASP SAMM est structuré en cinq fonctions métier, chacune comprenant des flux spécifiques couvrant différents aspects de la sécurité logicielle. Pour chaque flux, le *framework* définit des exigences à satisfaire afin d'atteindre différents niveaux de maturité, guidant ainsi les organisations dans le renforcement progressif de leurs pratiques de sécurité.

Choisir un niveau de maturité équivalent pour une recommandation :




- Identifier l'objectif principal de la recommandation (contrôle d'accès, pratiques de codage sécurisées, conformité, ...)
- Associer la recommandation à un flux et une fonction métier SAMM pertinents
- Attribuer un niveau de maturité en s'appuyant sur les tableaux récapitulatifs SAMM pour chaque fonction métier et flux, en tenant compte des exigences et activités spécifiques associées à l'atteinte de chaque niveau, ainsi que de la complexité de mise en œuvre de la recommandation.








OWASP SAMM Framework - Governance

Strategy & Metrics

Stream A: create & promote	Stream B: Measure & Improve
Maturity level 1  <p>Identify objectives and means of measuring effectiveness of the security program</p>	
Identify organisation drivers as they relate to the organisation's risk tolerance	Define metrics with insight into the effectiveness and efficiency of the Application Security Program
Maturity level 2  <p>Establish a unified strategic roadmap for software security within the organisation.</p>	
Publish a unified strategy for application security	Set targets and KPI's for measuring the program effectiveness
Maturity level 3  <p>Align security efforts with the relevant organisational indicators and asset values</p>	
Align the application security program to support the organisation's growth	Influence the strategy based on the metrics and organisational needs

Policy & Compliance

Stream A: Policy & Standards	Stream B: Compliance Management
Maturity level 1  <p>Identify and document governance and compliance drivers relevant to the organisation</p>	
Determine a security baseline representing organisation's policies and standards	Identify 3rd-party compliance drivers and requirements and map to existing policies and standards
Maturity level 2  <p>Establish application-specific security and compliance baseline</p>	
Develop security requirements applicable to all applications	Publish compliance-specific application requirements and test guidance
Maturity level 3  <p>Measure adherence to policies, standards, and 3rd-party requirements.</p>	
Measure and report on the status of individual application's adherence to policies and standards	Measure and report on individual application's compliance with 3rd party requirements

Education & Guidance

Stream A: Training & Awareness	Stream B: Organisation & Culture
Maturity level 1  <p>Offer staff access to resources around the topics of secure development and deployment</p>	
Provide security awareness training for all personnel involved in software development	Identify a "Security Champion" within each development team
Maturity level 2  <p>Educate all personnel in the software lifecycle with technology and role-specific guidance on secure development</p>	
Offer technology and role-specific guidance, including security nuances of each language and platform	Develop a secure software centre of excellence promoting thought leadership among developers and architects
Maturity level 3  <p>Develop in-house training programs facilitated by developers across different teams.</p>	
Standardised in-house guidance around the organisation's secure software development standards	Build a secure software community including all organisation people involved in software security



OWASP SAMM Framework - Design

Threat assessment

Stream A: Application Risk Profile	Stream B: Threat Modeling
Maturity level 1 ● ● ● Best-effort identification of high-level threats to the organisation and individual projects	
A basic assessment of the application risk is performed to understand likelihood and impact of an attack	Perform best-effort, risk-based threat modelling using brainstorming and existing diagrams with simple threat checklists
Maturity level 2 ● ● ● Standardisation and enterprise-wide analysis of software-related threats within the organisation	
Understand the risk for all applications in the organisation by centralising the risk profile inventory for stakeholders	Standardise threat modelling training, processes, and tools to scale across the organisation
Maturity level 3 ● ● ● Proactive improvement of threat coverage throughout the organisation	
Periodically review application risk profiles at regular intervals to ensure accuracy and reflect current state	Continuously optimisation and automation of your threat modeling methodology

Security requirements

Stream A: Software Requirements	Stream B: Supplier Security
Maturity level 1 ● ● ● Consider security explicitly during the software requirements process	
High-level application security objectives are mapped to functional requirements	Evaluate the supplier based on organisational security requirements
Maturity level 2 ● ● ● Increase granularity of security requirements derived from business logic and known risks.	
Structured security requirements are available and utilised by developer teams	Build security into supplier agreements in order to ensure compliance with organisational requirements
Maturity level 3 ● ● ● Mandate security requirements process for all software projects and third-party dependencies.	
Build a requirements framework for product teams to utilise	Ensure proper security coverage for external suppliers by providing clear objectives

Security architecture

Stream A: Architecture Design	Stream B: Technology Management
Maturity level 1 ● ● ● Insert consideration of proactive security guidance into the software design process	
Teams are trained on the use of basic security principles during design	Elicit technologies, frameworks and integrations within the overall solution to identify risk
Maturity level 2 ● ● ● Direct the software design process toward known secure services and secure-by-default designs	
Establish common design patterns and security solutions for adoption	Standardise technologies and frameworks to be used throughout the different applications
Maturity level 3 ● ● ● Formally control the software design process and validate utilisation of secure components	
Reference architectures are utilised and continuously evaluated for adoption and appropriateness	Impose the use of standard technologies on all software development



OWASP SAMM Framework - Implementation

Secure build

Stream A: Build Process	Stream B: Software Dependencies
Maturity level 1 <p>Build process is repeatable and consistent</p>	
Create a formal definition of the build process so that it becomes consistent and repeatable	Create records with Bill of Materials of your applications and opportunistically analyse these
Maturity level 2 <p>Build process is optimised and fully integrated into the workflow</p>	
Automate your build pipeline and secure the used tooling. Add security checks in the build pipeline	Evaluate used dependencies and ensure timely reaction to situations posing risk to your applications
Maturity level 3 <p>Build process helps prevent known defects from entering the production environment</p>	
Define mandatory security checks in the build process and ensure that building noncompliant artifacts fails	Analyse used dependencies for security issues in a comparable way to your own code

Secure deployment

Stream A: Deployment Process	Stream B: Secret Management
Maturity level 1 <p>Deployment processes are fully documented</p>	
Formalise the deployment process and secure the used tooling and processes	Introduce basic protection measures to limit access to your production secrets
Maturity level 2 <p>Deployment processes include security verification milestones</p>	
Automate the deployment process over all stages and introduce sensible security verification tests	Inject secrets dynamically during deployment process from hardened storages and audit all human access to them
Maturity level 3 <p>Deployment process is fully automated and incorporates automated verification of all critical milestones</p>	
Automatically verify integrity of all deployed software, independently on whether it's internally or externally developed	Improve the lifecycle of application secrets by regularly generating them and by ensuring proper use




Defect management

Stream A: Defect Tracking	Stream B: Metrics & Feedback
Maturity level 1 <p>All defects are tracked within each project</p>	
Introduce a structured tracking of security defects and make knowledgeable decisions based on this information	Regularly go over previously recorded security defects and derive quick wins from basic metrics
Maturity level 2 <p>Defect tracking used to influence the deployment process</p>	
Rate all security defects over the whole organisation consistently and define SLAs for particular severity classes	Collect standardised defect management metrics and use these also for prioritisation of centrally driven initiatives
Maturity level 3 <p>Defect tracking across multiple components is used to help reduce the number of new defects</p>	
Enforce the predefined SLAs and integrate your defect management system with other relevant tooling	Continuously improve your security defect management metrics and correlate it with other sources






OWASP SAMM Framework - Verification




Architecture assessment

Stream A: Architecture Validation	Stream B: Architecture Mitigation
Maturity level 1  Review the architecture to ensure baseline mitigations are in place for typical risks	
Identify application and infrastructure architecture components and review for basic security provisioning	Ad-hoc review of the architecture for unmitigated security threats
Maturity level 2  Review the complete provision of security mechanisms in the architecture	
Validate the architecture security mechanisms	Analyse the architecture for known threats
Maturity level 3  Review the architecture effectiveness and feedback results to improve the security architecture	
Review of the architecture components' effectiveness	Feed the architecture review results back into the enterprise architecture, organisation design principles and patterns, security solutions and reference architectures

Requirements-driven testing

Stream A: Control Verification	Stream B: Misuse/Abuse Testing
Maturity level 1  Opportunistically find basic vulnerabilities and other security issues	
Test for software security controls	Perform security fuzzing testing
Maturity level 2  Perform implementation review to discover application-specific risks against the security requirements	
Derive test cases from known security requirements	Create and test abuse cases and business logic flaw test
Maturity level 3  Maintain the application security level after bug fixes, changes or during maintenance	
Perform regression testing (with security unit tests)	Denial of service and security stress testing

Security testing

Stream A: Scalable Baseline	Stream B: Deep Understanding
Maturity level 1  Perform security testing (both manual and tool based) to discover security defects	
Utilise automated security testing tools	Perform manual security testing of high-risk components
Maturity level 2  Make security testing during development more complete and efficient through automation complemented with regular manual security penetration tests	
Employ application-specific security testing automation	Conduct manual penetration testing
Maturity level 3  Embed security testing as part of the development and deployment processes	
Integrate automated security testing into the build and deploy process	Integrate security testing into development process



OWASP SAMM Framework - Operations

Incident management

Stream A: Incident Detection	Stream B: Incident Response
Maturity level 1 Best-effort incident detection and handling	
Use available log data to perform best-effort detection of possible security incidents	Identify roles and responsibilities for incident response
Maturity level 2 Formal incident management process in place	
Follow an established, well-documented process for incident detection, with emphasis on automated log evaluation	Establish a formal incident response process and ensure staff are properly trained in performing their roles
Maturity level 3 Mature incident management	
Use a proactively managed process for detection of incidents	Employ a dedicated, well-trained incident response team

Environment management

Stream A: Configuration Hardening	Stream B: Patching & Updating
Maturity level 1 Best-effort patching and hardening	
Perform best-effort hardening of configurations, based on readily available information	Perform best-effort patching of system and application components
Maturity level 2 Formal process with baselines in place	
Perform consistent hardening of configurations, following established baselines and guidance	Perform regular patching of system and application components, across the full stack. Ensure timely delivery of patches to customers
Maturity level 3 Conformity with continuously improving process enforced	
Actively monitor configurations for nonconformance to baselines, and handle detected occurrences as security defects.	Actively monitor update status and manage missing patches as security defects. Proactively obtain vulnerability and update information for components

Operational management

Stream A: Data Protection	Stream B: Legacy Management
Maturity level 1 Foundational Practices	
Implement basic data protection practices	Decommission unused applications and services as identified. Manage customer upgrades/migrations individually
Maturity level 2 Managed, Responsive Processes	
Develop data catalogue and establish data protection policy	Develop repeatable decommissioning processes for unused systems/services, and for migration from legacy dependencies. Manage legacy migration roadmaps for customers.
Maturity level 3 Active Monitoring and Response	
Automate detection of policy noncompliance, and audit compliance periodically. Regularly review and update to data catalogue and data protection policy.	Proactively manage migration roadmaps, for both unsupported end-of-life dependencies, and legacy versions of delivered software.



Framework SLSA



Le framework SLSA (Supply-chain Levels for Software Artifacts) renforce la sécurité de la chaîne d'approvisionnement logicielle à travers quatre niveaux, exigeant une application cohérente de chaque niveau sur le contrôle du code source, le *build* et le déploiement pour atteindre le niveau global correspondant. En d'autres termes, un niveau SLSA ne peut être atteint que si ce même niveau est satisfait pour chacune des dimensions — contrôle du code source, *build* et déploiement — à l'exception du niveau 1, qui correspond aux mesures de sécurité les plus fondamentales.

Identifier un niveau de sécurité équivalent pour une recommandation :

- Identifier l'aspect de la chaîne d'approvisionnement logicielle concerné par la recommandation (contrôle du code source, *build*, déploiement).
- Associer la recommandation à un niveau de sécurité SLSA : se référer aux tableaux fournis qui définissent les critères pour différents niveaux entre contrôle de version, compilation et déploiement.

Overall levels of security

	Source code control	Build	Deployment
Level 1: basic integrity	Use a version control system to track changes	The build process must be documented and produce provenance	Basic checks to ensure consistency between build artifacts and deployed software
Level 2: defence against tampering	Ensure the change history is verified and immutable.	Use a hosted build platform with signed provenance	Ensure that deployment artifacts are signed and verified
Level 3: increased security	Require code reviews by multiple trusted persons before changes are submitted	The build environment must be isolated from the source control system, and the provenance must be authenticated	Deployment artifacts must be verified and come from trusted build processes
Level 4: highest security	All changes must be rigorously reviewed, and the system must provide high confidence in the provenance of the source	The build process must be hermetic (isolated from external dependencies) and reproducible	Deployment processes must include stringent verification of artifact integrity and ensure that only trusted sources are used



Source control levels

	Criteria
Level 1: version controlled	<ul style="list-style-type: none"> • Use of a modern version control system • Every change to the source code is tracked • Basic attribution for changes (who made the change and when).
Level 2: verified history	<ul style="list-style-type: none"> • Multi-factor authentication (MFA) for all user accounts • Immutable change history (cannot be altered or deleted) • Timestamps verified by the source control platform.
Level 3: modifications authorisation	<ul style="list-style-type: none"> • Mandatory code reviews by at least two trusted individuals • Enforced code review policy (informed reviews and context-specific approvals) • Use of MFA for accounts performing reviews

Build levels

	Criteria
Level 1: documentation and provenance	<ul style="list-style-type: none"> • Documentation of the build process • Generation of provenance (metadata) that describes how the software was built.
Level 2: hosted build and signed provenance	<ul style="list-style-type: none"> • Builds are executed on a hosted build platform • Provenance is signed by the build service to ensure authenticity
Level 3: hardened builds	<ul style="list-style-type: none"> • Builds are isolated from the source control environment • Build systems generate tamper-resistant provenance • Builds are reproducible and hermetic (isolated from external dependencies).

Deployment levels

	Criteria
Level 1: basic packaging	<ul style="list-style-type: none"> • Simple checks to ensure that the deployed software matches the built artifact
Level 2: signed artifacts	<ul style="list-style-type: none"> • Artifacts (e.g., binaries, containers) are cryptographically signed • Verification processes in place to check signatures before deployment
Level 3: verified and trusted deployment	<ul style="list-style-type: none"> • Deployment processes check that artifacts are signed and verified • Only trusted sources and build pipelines are allowed to deploy artifacts • Continuous monitoring of deployed software for integrity

Remerciements

L'ANSSI a été **accompagnée par Wavestone** dans la réalisation de cette étude de marché et remercie l'ensemble des consultants impliqués.

L'ANSSI et Wavestone remercient les **9 bénéficiaires** et les **13 éditeurs rencontrés** lors de cette étude pour leur disponibilité et le partage de leurs travaux : [SonarSource](#), [Codacy](#), [Aikido Security](#), [Bytesafe](#), [Gitguardian](#), [ImmuniWeb](#), [Hackuity](#), [BlackDuck](#), [Snyk](#), [CAST Software](#), [OWASP](#), [Mend](#) et [Cybeats](#).



Pour toute question portant sur cette étude, n'hésitez pas à contacter la Division Industrie et Technologies de l'ANSSI à l'adresse suivante : industries@ssi.gouv.fr