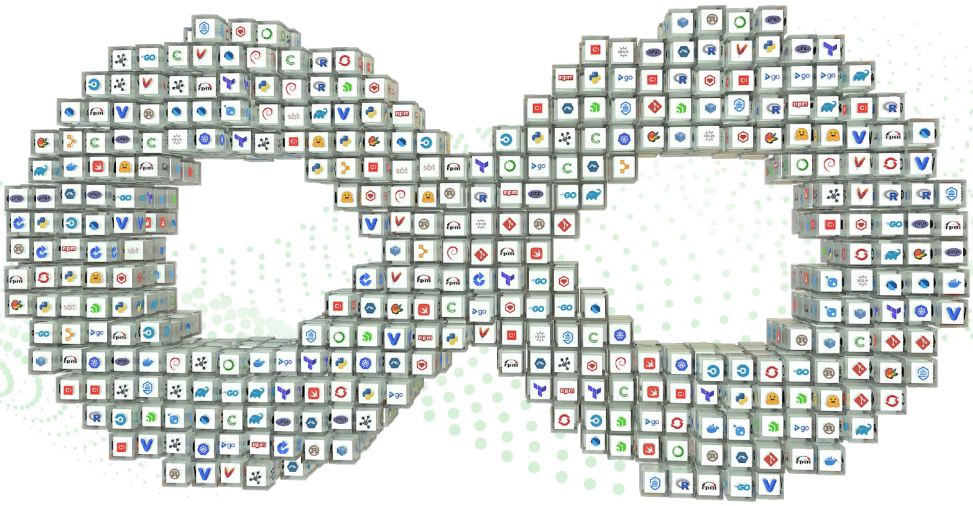
L’état de la chaîne d’approvisionnement logicielle en 2025

L’élargissement du paysage des menaces met en péril l’intégrité des logiciels

Table des matières

Introduction



La gestion et la sécurisation de l'ensemble de la chaîne d'approvisionnement logicielle sont essentielles pour fournir des versions de logiciels fiables. Cependant, cela est souvent plus facile à dire qu'à faire. Avec plus de 15 ans d'accompagnement des équipes de développement et de sécurité, une équipe de recherche dédiée et un fort accent sur la sécurité logicielle, JFrog possède une compréhension approfondie des menaces et enjeux rencontrés par les entreprises actuellement. Dans un monde post-AI, ces défis ne font que s'accélérer, laissant la plupart des équipes DevSecOps devant une impasse : comment faire face à tous ces changements ?

Ce rapport combine les données d'utilisation de JFrog provenant de millions d'utilisateurs, l'analyse de CVE par l'équipe de recherche en sécurité de JFrog et les données d'enquêtes tierces commandées auprès de 1 400 professionnels de la sécurité, du développement et des opérations afin de répondre à cette question primordiale. L'analyse qui en résulte permet de situer le contexte de la chaîne d'approvisionnement et du développement de logiciels dans son ensemble, de révéler où se situent les risques persistants et les nouveaux risques, et d'explorer ce qu'il faut faire pour sécuriser votre chaîne d'approvisionnement logicielle en 2025.

Nous espérons que ce rapport vous sera utile et nous vous invitons à nous faire part de vos commentaires à l'adresse suivante : data_report@jfrog.com.

Note de synthèse

L'évolution fulgurante de la chaîne d'approvisionnement logicielle risque de confronter les organisations à des menaces inédites et de plus en plus difficiles à maîtriser. Lorsqu'il s'agit d'atténuer les risques tout au long de la chaîne d'approvisionnement, le « faire plus » n'est pas nécessairement la meilleure approche. Adopter la philosophie « travailler plus intelligemment, pas plus durement », en misant sur la simplification des outils et des processus, permettra aux organisations d'évoluer rapidement, d'adopter les technologies émergentes et de prendre l'avantage sur la concurrence.

Votre chaîne logistique logicielle s'est agrandie, accélérée et complexifiée

La croissance de l'écosystème open source ne montre aucun signe de ralentissement, et les organisations désireuses d'innover se hâtent de tirer parti des technologies les plus récentes.

- Deux tiers des organisations (64%) déclarent utiliser 7 langages de programmation ou plus. 44 % en utilisent 10 ou plus. Cela représente une hausse par rapport à l'année précédente, où les taux étaient respectivement de 53 % et 31 %.
- Les dépôts publics continuent de croître. À noter qu'en 2024, Docker Hub s'est doté de 1,9 million d'images supplémentaires et Hugging Face 1 million.
- L'organisation type introduit 458 nouveaux packages par an. Cela représente 38 nouveaux packages par mois, en moyenne, et varie en fonction du nombre de développeurs.

Plus de risques, moins de clarté

Si la compréhension de l'impact potentiel d'une CVE reste une entreprise complexe, il ne s'agit que de la partie émergée de l'iceberg des risques encourus.

- L'énorme retard accumulé par la National Vulnerability Database (NVD) n'a pas empêché la découverte de nouvelles vulnérabilités. Plus de 33 000 nouvelles CVE ont été signalées en 2024, soit une augmentation de 27 % en glissement annuel.
- L'équipe de recherche en sécurité de JFrog a détecté 25 229 secrets/tokens exposés dans les registres publics (en hausse de 64 % par rapport à l'année précédente), dont 6 790 étaient actifs.
- Lors d'une analyse approfondie de 183 CVE notables, l'équipe de recherche en sécurité de JFrog a découvert que 63 d'entre elles ne seraient jamais exploitables dans les applications analysées des clients de JFrog Cloud.

Face aux risques de sécurité, il ne faut surtout pas ignorer les principes essentiels

Les organisations ont adopté différents niveaux de référentiels de sécurité et recourent à davantage d'outils, mais des pratiques clés échappent toujours à leur attention.

- 71 % des personnes interrogées indiquent que leur organisation permet aux développeurs de télécharger des packages directement à partir d'Internet.
- 73 % des organisations utilisent 7 solutions de sécurité ou plus. 49 % en utilisent 10 ou plus. Ces chiffres sont en hausse par rapport à l'année dernière (47 % et 33 %).
- Moins de la moitié des personnes interrogées (43 %) indiquent que leur organisation analyse le code et le niveau binaire.
- 40 % des personnes interrogées ne disposent pas d'une visibilité totale sur la provenance des logiciels en production.

L'adoption de l'IA passe à la vitesse supérieure

Les équipes disposent de plus d'options que jamais pour mettre en production des services d'IA, mais cela pose de nouveaux problèmes que les organisations doivent résoudre.

- Hugging Face a vu plus d'un million de nouveaux modèles et ensembles de données publiés cette année, mais cette croissance s'est traduite par une multiplication par 6,5 des modèles malveillants.
- Les équipes se tournent vers des modèles hébergés (64 %), mais près de la moitié des organisations hébergent également elles-mêmes des modèles d'une manière ou d'une autre, qu'ils soient propriétaires ou open source.
- 37 % des organisations s'appuient actuellement sur des efforts manuels pour établir et maintenir une liste de modèles approuvés afin de régir l'utilisation des artefacts des modèles.

Que contient votre chaîne d’approvisionnement logicielle ?

La chaîne d’approvisionnement logicielle moderne est globale et étendue, intégrant de multiples technologies et sources, avec des millions de nouveaux packages et bibliothèques ajoutés chaque année aux écosystèmes technologiques les plus populaires. Les organisations de développement de logiciels exploitent aujourd’hui un nombre sans précédent de langages et d’écosystèmes de packages correspondants. Si les technologies traditionnelles restent largement utilisées, l’innovation dans les écosystèmes open source établis et émergents présente des opportunités et des risques qui seront examinés plus en détail dans le présent rapport.

Nombre de langages de programmation utilisés dans les organisations de développement

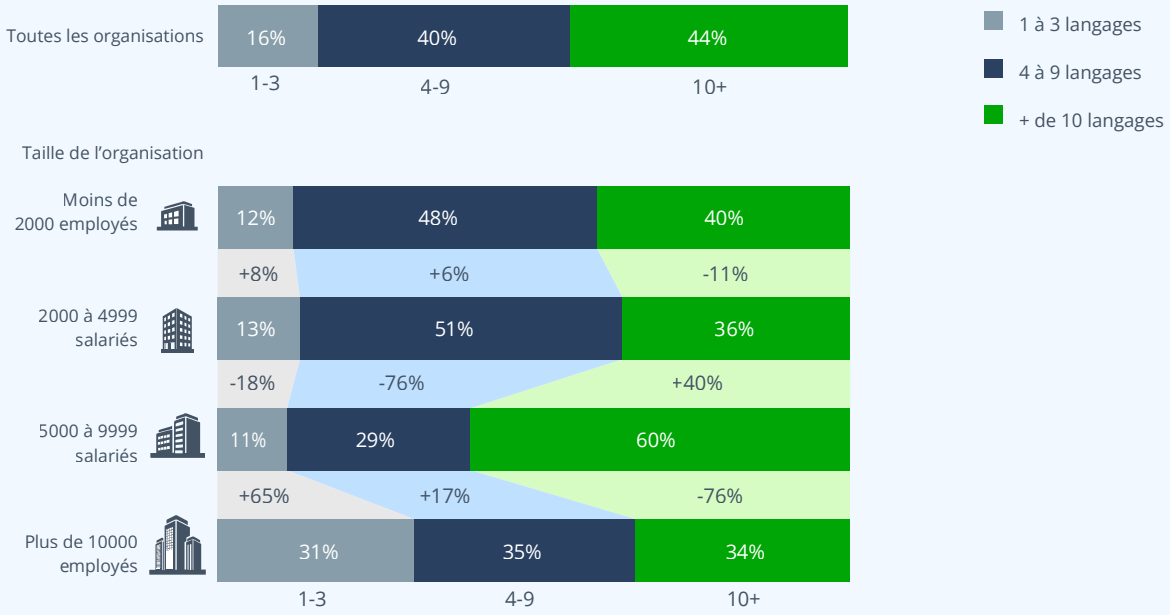


Figure 1.1. Au sein de votre équipe de développement, combien de langages de programmation sont utilisés ? (Enquête commanditée, 2024)

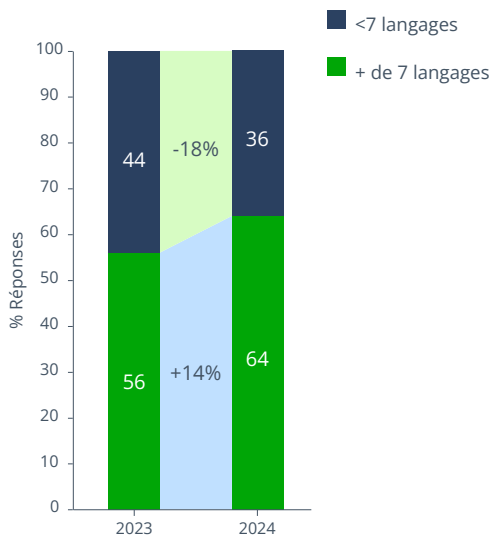


Figure 1.2

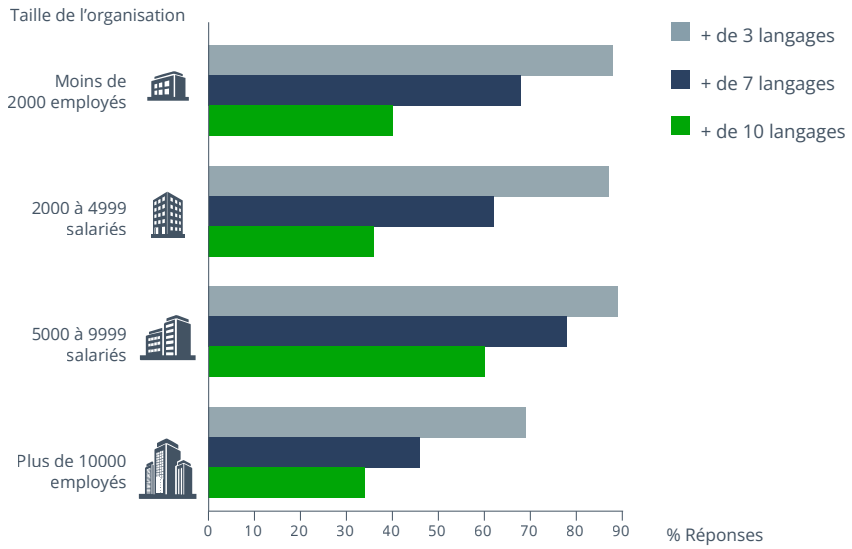


Figure 1.3

Près de deux tiers des professionnels de la technologie (64 %) déclarent que leur entreprise utilise 7 langages de programmation ou plus. L'année dernière, un peu plus de la moitié des personnes interrogées (56 %) déclaraient la même chose. Cette augmentation reflète l'accroissement général de la complexité que nous observons dans la chaîne d'approvisionnement logicielle.

Plus la taille de l'organisation augmente, plus le nombre de langages utilisés tend à augmenter, ce qui est une tendance prévisible. Toutefois, dès que la taille de l'entreprise atteint 10 000 employés ou plus, le nombre de langages utilisés diminue. Cette diminution traduit vraisemblablement une étape où les organisations prennent conscience qu'il est temps d'adopter une approche plus proactive de la gestion du développement et de normaliser l'emploi de technologies spécifiques pour éviter l'éparpillement. Il se peut aussi que les organisations de plus grande taille se contentent de maintenir des applications legacy existantes, ce qui implique moins de nouveaux projets nécessitant l'intégration de technologies supplémentaires.

Nouveaux packages par an et par type de package

Figure 2. Nombre de nouveaux packages par an, affiché par type de package (base de données JFrog Catalog, 2024)

D'après les données du catalogue JFrog analysant les registres publics, Docker Hub reste, de loin, l'écosystème recevant le plus de contributions. Bien que le rythme de croissance ait quelque peu ralenti en 2024 par rapport à la croissance explosive de 2023, un nombre incroyable de 2 millions de nouveaux packages ont été introduits en 2024. Une partie de cette croissance à partir de 2023 et jusqu'en 2024, et peut-être le ralentissement du rythme, est due à la campagne malveillante découverte par l'équipe de recherche en sécurité de JFrog, que nous aborderons plus loin dans ce rapport.

> Après un examen préliminaire, l'équipe de recherche en sécurité de JFrog n'a pas trouvé d'activité malveillante liée au pic de 4x dans Debian, ce qui indique que cette croissance est probablement due à d'autres raisons.

La recherche de cette année comprend également une analyse de RubyGems, Debian, Alpine, Conan et Hugging Face. La croissance exponentielle que nous observons avec Hugging Face illustre l'intérêt généralisé et croissant pour l'IA/ML, à la fois en termes de développement et d'incorporation des capacités de l'IA/ML dans les applications d'entreprise. Nous nous attendons à ce que la croissance de l'écosystème se poursuive cette année et au-delà.

Les technologies les plus utilisées par les entreprises

Type de packages	Requêtes*	Nombre de dépôts	Artefacts
Maven	33.52%	104,955	2,567,881,564
npm	30.45%	48,549	674,010,130
Docker	15.45%	112,366	2,264,459,098
YUM	2.68%	14,669	20,785,724
PyPI	2.68%	22,352	66,838,230
Helm	1.61%	26,125	13,231,209
Nuget	1.45%	28,497	131,164,087
Debian	1.35%	8,184	8,066,185
Conan	1.33%	3,420	143,404,846
Gradle	0.99%	9,073	102,198,342
RubyGems	0.93%	3,736	46,728,889
Go	0.75%	9,034	16,511,299
OCI	0.47%	862	8,662,480
Cargo	0.13%	1,261	526,851
Sbt	0.12%	2,239	14,908,497
Helm OCI	0.07%	1,633	201,440
Ivy	0.06%	2,283	31,786,069
Composer	0.05%	2,413	614,957
Terraform	0.03%	3,566	675,684
Opkg	0.02%	529	33,812,836
Conda	0.02%	2,168	1,538,832
P2	0.02%	316	1,010,616
Pub	0.01%	363	166,878
Swift	0.01%	524	1,345,299
Alpine	0.01%	1,550	111,231
Cocoapods	<0.01%	1,400	2,973,045
Cran	<0.01%	2,403	816,170
VCS	<0.01%	273	1,692
Chef	<0.01%	1,530	150,462
Vagrant	<0.01%	680	7,326
Terraform Backend	<0.01%	2,307	395,004
Bower	<0.01%	985	44,161
Ansible	<0.01%	107	4,470
Puppet	<0.01%	1,530	17,758
Hugging Face	<0.01%	551	12,638

*% du total des requêtes sur 57 milliards de requêtes au T4

Cette année, nous avons réalisé une analyse à la fin du quatrième trimestre afin d'obtenir une vision plus précise des technologies les plus populaires parmi les plus de 35 types de technologies pris en charge par JFrog. Si les écosystèmes technologiques bien établis, tels que npm, Docker et Maven, continuent de prévaloir, YUM et Cargo ont enregistré des gains de popularité notables.

Au cours des dernières années, nous avons constaté que la popularité de Cargo n'a cessé de croître, en particulier parce que les entités gouvernementales font pression pour un développement plus sûr de la mémoire. Il reste à voir si la popularité de Rust plafonnera ou atteindra les niveaux d'utilisation et d'adoption de langages plus établis comme Java.

Il convient également de souligner l'importance de l'utilisation d'OCI et de Helm OCI. JFrog a introduit des référentiels dédiés à l'OCI au début de l'année 2024 et beaucoup de nos clients en profitent déjà. Cela témoigne d'une préférence croissante pour des standards ouverts en matière de conteneurs et d'autres écosystèmes technologiques, ce qui nous a amenés à élargir nos dépôts Terraform afin de prendre en charge nativement OpenTofu.

L'utilisation de technologies communes diffère selon les secteurs :

- **Les entreprises des secteurs de l'automobile et de l'IoT** utilisent Maven (applications back-end), npm (applications front-end), Conan (dispositifs embarqués), Docker, PyPI (pour l'IA/ML), et regroupent souvent plusieurs de ces technologies dans des packages génériques (images tar/zip).
- **Les entreprises d'IA/ML et de robotique** s'appuient sur PyPI, sur des modèles ML tirés de dépôts publics tels que Hugging Face et Tensorflow, et stockent ces modèles dans des conteneurs ou des packages génériques (tar/zips). Elles peuvent également adopter des dépôts natifs tels que Hugging Face ou JFrog's Machine Learning Repository* pour leurs modèles.
- **Les institutions des secteurs de l'assurance, de la finance et du commerce de détail** adoptent une combinaison de technologies telles que Maven, npm et Docker. Avec la montée en puissance de l'IA et du Machine Learning, elles commencent également à utiliser PyPI et des modèles ML pour proposer des offres plus avancées et rester compétitives.

*JFrog's Machine Learning Repository a été introduit en janvier 2025 et n'est pas inclus dans les données de ce rapport.

Figure 3. Technologies utilisées, nombres d'actions, nombre de dépôts et taille totale des artefacts stockés pour chacune (base de données JFrog, 2024)

Bibliothèques populaires

Rank	Docker	Maven	PyPI	npm
1	library/alpine	org.slf4j:slf4j-api	urllib3	@types/node
2	library/node	commons-io:cozmons-io	requests	semver
3	library/python	commons-codec:commons-codec	certifi	minimatch
4	library/nginx	org.ow2.asm:asm	charset-normalizer	glob
5	library/redis	com.fasterxml.jackson.core:jackson-core	setuptools	electron-to-chromium
6	library/busybox	com.google.guava:guava	idna	lru-cache
7	library/postgres	com.fasterxml.jackson.core:jackson-databind	packaging	caniuse-lite
8	library/ubuntu	com.fasterxml.jackson. core:jackson-annotations	typing-extensions	acorn
9	library/openjdk	org.apache.commons:commons-compress	wheel	debug
10	library/debian	org.apache.commons:commons-lang3	PyYAML	@babel/parser
11	grafana/grafana	org.codehaus.plexus:plexus-utils	python-dateutil	strip-ansi
12	library/golang	junit:junit	numpy	browserslist
13	library/hello-world	org.apache.httpcomponents:httpcore	click	@babel/types
14	library/maven	org.apache.httpcomponents:httpclient	MarkupSafe	tslib
15	library/docker	com.google.code.findbugs:jsr305	pytz	resolve
16	library/eclipse-temurin	com.google.errorprone:error_prone_annotations	cryptography	commander
17	curlimages/curl	commons-logging:commons-logging	cffi	qs
18	library/mongo	net.bytebuddy:byte-buddy	importlib-metadata	@babel/code-frame
19	library/centos	org.objenesis:objenesis	zipp	@babel/generator
20	library/amazoncorretto	org.apache.maven:maven-artifact	attrs	chalk

Figure 4. Les 20 packages les plus téléchargés pour Docker, Maven, PyPI, npm dans JFrog Cloud (SaaS) (base de données JFrog, 2024)

De nombreux registres publics proposent des mesures de téléchargement pour les packages qu'ils contiennent, mais ces mesures peuvent être trompeuses pour diverses raisons, notamment parce qu'elles sont influencées par des éléments tels que les clients qui récupèrent le package à chaque fois qu'un build est exécuté. À la place, notre étude détermine les bibliothèques réellement utilisées en se basant sur celles qui sont effectivement requises dans les environnements JFrog SaaS, qui est utilisé par des milliers de clients.

Pour Docker, il n'est pas surprenant que les 20 premières images comprennent les systèmes d'exploitation les plus populaires et les principaux langages de développement, probablement utilisés comme images parentes. Il est encourageant de constater que toutes les images, sauf une, sont soit des images officielles de Docker, soit des images fournies par un éditeur vérifié, ce qui indique que l'on veille à ce que ces images soient régulièrement mises à jour. Fait notable, l'image officielle Docker helloworld fait partie du peloton de tête, ce qui laisse supposer une abondance de démonstrations, de projets de preuve de concept et de développeurs qui découvrent Docker, alors que cette technologie de conteneurisation devient un standard dans la distribution des logiciels.

En ce qui concerne les packages Maven, PyPI et npm, il n'y a pas eu de surprise dans le top 20 des packages les plus utilisés par les organisations. Cela dit, il est difficile de savoir si ces packages sont intégrés directement, s'ils sont choisis explicitement par les développeurs de logiciels ou s'ils sont intégrés sous forme de dépendances ou même de dépendances transitives.

À titre d'exemple, Apache Commons Compress se positionne au 9e rang dans le classement de popularité. Si vous examinez cette bibliothèque de plus près, vous constaterez qu'elle dépend directement d'Apache Commons IO, Apache Commons Codec, ASM et Apache Commons Lang, à la 2e, 3e, 4e et 10e place, respectivement. Cela souligne l'importance de maintenir un inventaire à jour des artefacts logiciels inclus dans une application, souvent sous la forme d'un SBOM, afin d'évaluer chaque ingrédient individuel et d'avoir une meilleure compréhension du rayon d'action en cas de vulnérabilité, de compromission ou de disparition d'un composant spécifique dans votre chaîne d'approvisionnement logicielle.

Rythme d'introduction de nouveaux packages OSS au sein d'une organisation

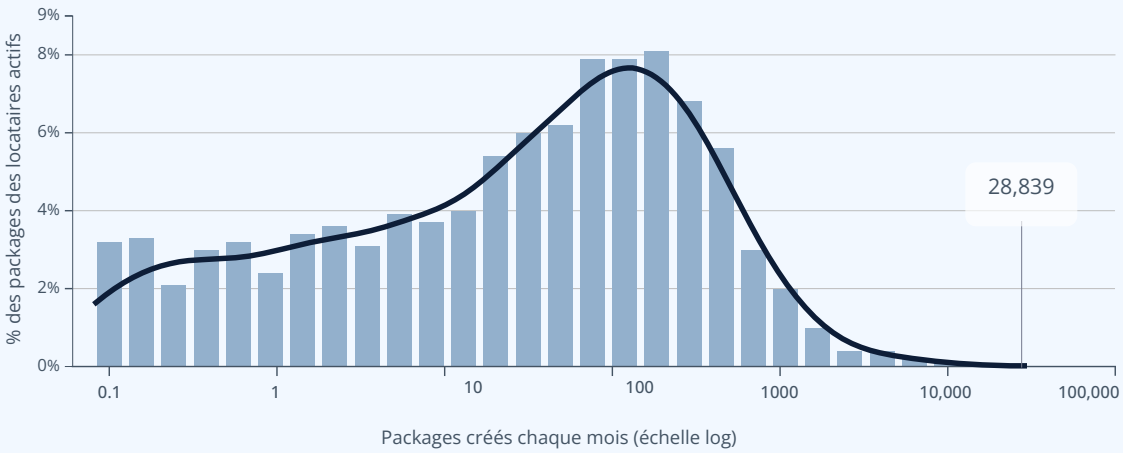


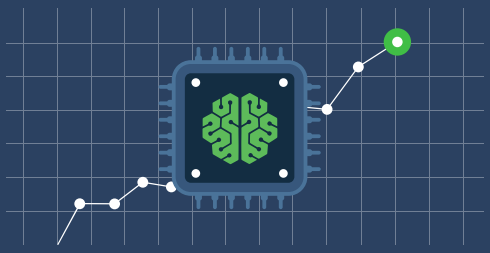
Figure 5. Répartition des nouveaux packages créés mensuellement pour les locataires actifs en 2024 (base de données JFrog, 2024)

En 2024, les organisations utilisant JFrog Cloud, l'offre SaaS cloud native de JFrog, ont introduit plus de sept millions de nouveaux packages dans leur chaîne d'approvisionnement logicielle.

Pour une organisation moyenne, cela représente environ 2 000 packages tout au long de l'année, mais ce chiffre est gonflé par quelques utilisateurs très importants. L'organisation la plus importante a reçu 346 000 nouveaux packages au cours de l'année, tandis que l'organisation médiane en a reçu 231, ce qui est beaucoup plus facile à gérer.

Si l'on exclut les organisations qui n'ont pas introduit de packages, le nombre médian de packages passe à 458, soit 38 nouveaux packages par mois. Sur la base des données, ce chiffre est probablement le plus représentatif d'une organisation type. Même à un rythme d'un peu plus d'un nouveau package par jour, les organisations sont confrontées à des défis importants en ce qui concerne la manière dont elles prennent en compte et gèrent la sécurité, le risque opérationnel et la conformité aux licences de ce qui est introduit dans leur environnement.

Éléments clés



Explosion de l'IA

Nous assistons à une croissance exponentielle de la disponibilité des composants AI/ML, avec de plus en plus d'acteurs communautaires et d'entreprises qui s'impliquent en contribuant à l'écosystème (par ex., Nvidia qui lance NIM et NVLM). Les entreprises s'empressent d'ajouter des services d'IA à leurs produits, comme en témoignent les plus de 500 dépôts Hugging Face créés aujourd'hui par les utilisateurs de JFrog. Il est important d'avoir des politiques et des stratégies bien définies sur la façon dont vous consommez et sécurisez les modèles et les ensembles de données open source ; un sujet que nous explorerons plus en détail dans ce rapport.

La protection des applications et de leur développement est une priorité

Le gouvernement américain et d'autres entités politiques mondiales ont fait pression pour l'utilisation de langages et de cadres de développement plus sûrs. JFrog commence également à voir l'augmentation de l'utilisation de Rust/Cargo dans ses propres données, ce qui indique que les organisations peuvent réarchitecturer les applications ou commencer de nouveaux projets en se basant davantage sur la sécurité. De plus, la popularité de l'OCI peut sans doute s'expliquer, en partie, par l'inquiétude croissante des organisations quant au passage de technologies open source populaires vers des modèles privés, sous licence commerciale, ou soumis à des restrictions d'utilisation.

Risque multiplié par 10

Avec les deux tiers des organisations utilisant au moins 7 langages et près de la moitié en utilisant 10 ou plus, le risque pour les entreprises augmente de manière exponentielle, car elles doivent désormais garantir une chaîne de développement cohérente pour de nombreux langages, équipes et sources de menaces différentes. Chaque écosystème présente des vulnérabilités, des acteurs malveillants et des structures uniques qui doivent être pris en compte lors du développement afin de garantir la sécurité des applications déployées en production.

Un package par jour ? Tenir les attaquants à distance

Les entreprises qui évoluent rapidement introduisent un ou plusieurs nouveaux packages et versions par jour, ce qui nécessite des processus automatisés et améliorés pour garantir la sécurité de ces composants introduits dans leur chaîne d'approvisionnement logicielle. Alors que les organisations cherchent en permanence à accélérer leurs processus et à donner plus d'autonomie aux développeurs et aux équipes de sécurité pour trouver des solutions innovantes aux problématiques d'entreprises, le rythme d'intégration de nouveaux packages au sein des entreprises ne devrait que s'intensifier.

L’accélération des risques dans votre chaîne d’approvisionnement logicielle

Les organisations sont engagées dans une course contre les acteurs malveillants et doivent faire face à un ensemble de facteurs clés qui ne montrent aucun signe de ralentissement :

- CVE
- Packages malveillants
- Risques liés aux licences open source
- Risques opérationnels (packages mal gérés, fin de vie, etc.)
- Exposition des secrets
- Erreurs de configuration / Erreurs humaines

Dans l’ensemble, l’analyse montre que les outils utilisés actuellement par les développeurs et les professionnels de la sécurité sont utiles dans certains cas et nuisibles dans d’autres. Par exemple, les assistants de code IA, s’ils ne sont pas utilisés de manière appropriée, peuvent avoir un impact potentiellement négatif, en particulier dans le cas de fonctions mal configurées.

Toute comparaison annuelle des scores CVSS (Common Vulnerability Scoring System) et des informations CWE (Common Weakness Enumeration) présentée dans cette section sera biaisée cette année, en raison de la période de plusieurs mois durant laquelle la National Vulnerability Database (NVD) n’a pas pu analyser ni attribuer de propriétés aux nouvelles vulnérabilités (CVE), entraînant ainsi un important retard de traitement.

Ce retard de la NVD est un signe de prudence, qui met en lumière un problème persistant dans notre secteur d’activité. Comme le nombre de bibliothèques, et donc de CVE, continue d’augmenter, les organisations doivent réfléchir à la meilleure façon de gérer ce risque accru de manière durable. En outre, compte tenu de la situation politique actuelle aux États-Unis, l’intégrité et la viabilité future de la NVD et du National Institute of Standards and Technology (NIST) ne sont pas garanties.

Vulnérabilités trouvées dans une technologie ou un type de package donné

En 2024, les chercheurs en sécurité du monde entier ont divulgué près de 33 000 nouvelles CVE, soit une augmentation de 27 % par rapport à 2023, ce qui confirme la croissance annuelle des CVE découvertes. Bien que cela ne soit pas surprenant vu le nombre toujours croissant de nouveaux packages open source, la progression du nombre de CVE (27 % d'augmentation annuelle) dépasse celle des packages (24,5 %), ce qui constitue un indicateur à prendre au sérieux.

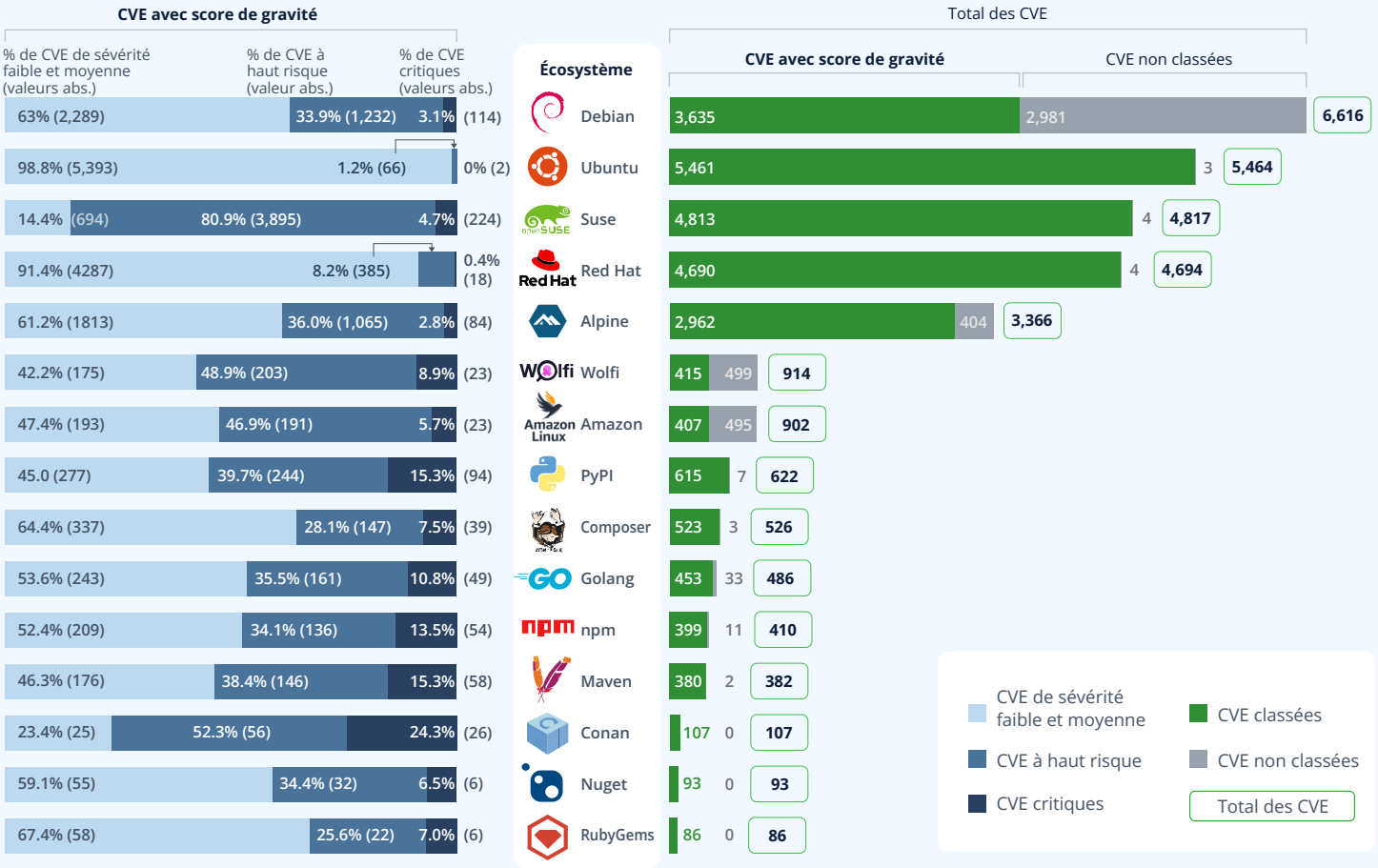


Figure 6.1. Nombre de CVE découvertes par type de package en 2024

Le premier point notable concerne la croissance marquée des CVE Debian d'une année sur l'autre. La hausse des CVE Debian n'est pas surprenante, étant donné que le nombre de packages ajoutés à l'écosystème a quadruplé en 2024. Heureusement, les CVE critiques et à haut risque sont peu nombreuses parmi les CVE Debian de 2024.

Comme pour les données de 2023, Maven, npm, PyPI et Conan (un nouvel ajout cette année) représentent le pourcentage le plus élevé de CVE critiques, même si le nombre total de CVE a diminué pour Maven et npm d'une année sur l'autre. Si l'on considère une vue d'ensemble de la base de données à la fin de l'année, le risque persistant met en évidence le niveau de risque important qui subsiste, notamment dans npm, Maven, et dans une moindre mesure PyPI.

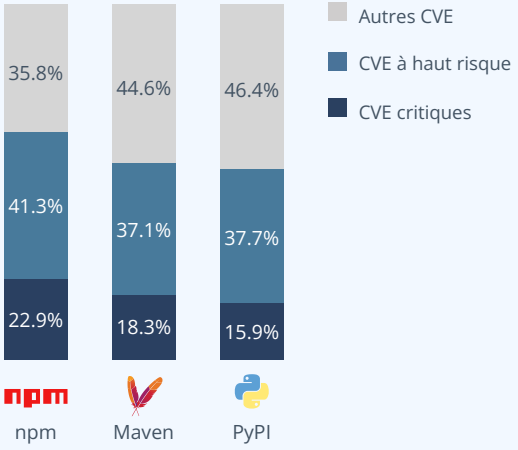


Figure 6.2. % de CVE critiques et à haut risque dans les écosystèmes populaires à la fin de l'année 2024

Total des packages supprimés et obsolètes

* Pas de données collectées pour l'année

Figure 7. Total des packages supprimés et obsolètes (base de données JFrog, 2024)

Les packages ne sont pas seulement ajoutés aux écosystèmes technologiques, ils sont aussi parfois supprimés. Le point le plus frappant parmi ces données est la hausse du nombre de packages Composer supprimés en 2024 par rapport à 2023. Un examen manuel effectué par l'équipe de recherche en sécurité de JFrog a permis de trouver que :

- La plupart des packages supprimés affichent un dépôt GitHub « non disponible », ce qui signifie qu'ils ont été supprimés par l'auteur ou rendus privés.
- Dans un cas observé, le package supprimé avait simplement été renommé ; il se pourrait donc que Packagist considère l'événement de renommage comme une « suppression ».
- Certains packages sont supprimés et marqués comme « abandonnés », bien que les critères d'abandon ne soient pas clairs.
- Il semble que Packagist ait probablement exécuté une nouvelle automatisation en 2024 qui a supprimé les packages avec des dépôts GitHub invalides.

Nos sources de données varient en ce qui concerne la déclaration des packages supprimés ; certaines fournissent cette information, d'autres non. Pour certains écosystèmes, nous analysons toutes les données disponibles et les comparons dans le temps, mais cette méthode ne tient pas compte des packages supprimés avant notre première collecte de données.

Dans d'autres cas, nous recevons des mises à jour périodiques qui incluent des informations depuis notre dernière exécution, et seuls certains écosystèmes signalent des suppressions lors de ces mises à jour. Nous ne disposons donc pas toujours d'informations complètes sur les packages supprimés.

Les types de vulnérabilités les plus courants

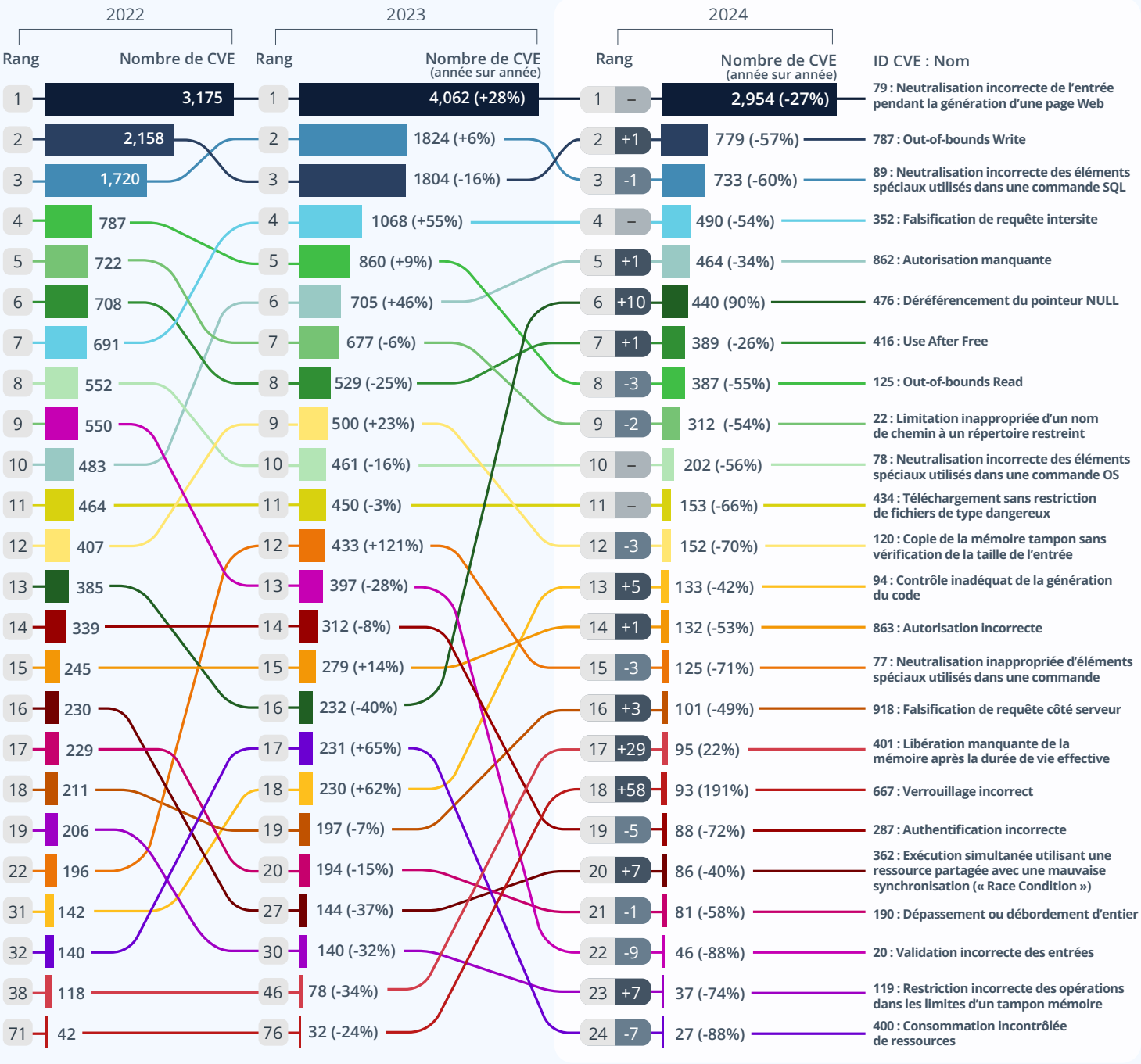


Figure 8. Vulnérabilités populaires divulguées en 2024 par rapport à 2023, 2022 et 2021

243 ID CWE uniques ont été attribués à des CVE en 2024, et les trois premiers restent constants par rapport à l'année précédente : Cross-site Scripting, Out-of-bounds Write et Injection SQL. Toutefois, trois nouvelles vulnérabilités ont fait leur apparition dans le top 20 des vulnérabilités les plus populaires, chacune ayant connu une croissance inhabituellement élevée :

401 : Libération manquante de la mémoire après la durée de vie effective

2024 #17 ↑ 2023 #46

362 : Exécution simultanée utilisant une ressource partagée avec une mauvaise synchronisation (« Race Condition »)

2024 #20 ↑ 2023 #28

Restriction incorrecte des opérations dans les limites d'un tampon mémoire.

2024 #23 ↑ 2023 #30

Pour traiter les trois CWE les plus courants (Cross-site Scripting, Out-of-bounds Write, et SQL Injection), qui peuvent être détectés par les outils SAST, nous recommandons aux équipes de scanner leur code source avec des outils SAST automatisés afin de prévenir de nouvelles vulnérabilités de ces types. En outre, Out-of-bounds Write est un problème propre aux langages de programmation bas niveau (c'est-à-dire non sécurisés en mémoire) tels que C ou C++. Ces problèmes peuvent être évités en passant à des langages de haut niveau, comme le suggère le gouvernement américain.

On notera que les tendances d'année en année liées aux types CWE peuvent être perturbées par des facteurs aléatoires ainsi que par des incidents exceptionnels. Par exemple, le retard accumulé par la NVD affecte presque certainement la représentation de la prévalence des CWE en 2024. Une fois que toutes les CVE auront été correctement répertoriées, ces chiffres changeront probablement.

L'examen d'une période plus large de 20 ans, par exemple, révélerait des tendances plus significatives, car la popularité des langages de bas niveau par rapport aux langages de haut niveau peut influer sur le nombre de vulnérabilités liées à la corruption de la mémoire par rapport aux problèmes de haut niveau ou liés au web. La fluctuation de la popularité de technologies spécifiques, chacune sujette à certains types de CWE, peut également affecter ces tendances.

Impacts des vulnérabilités communes pour les CVE les plus connues en 2024

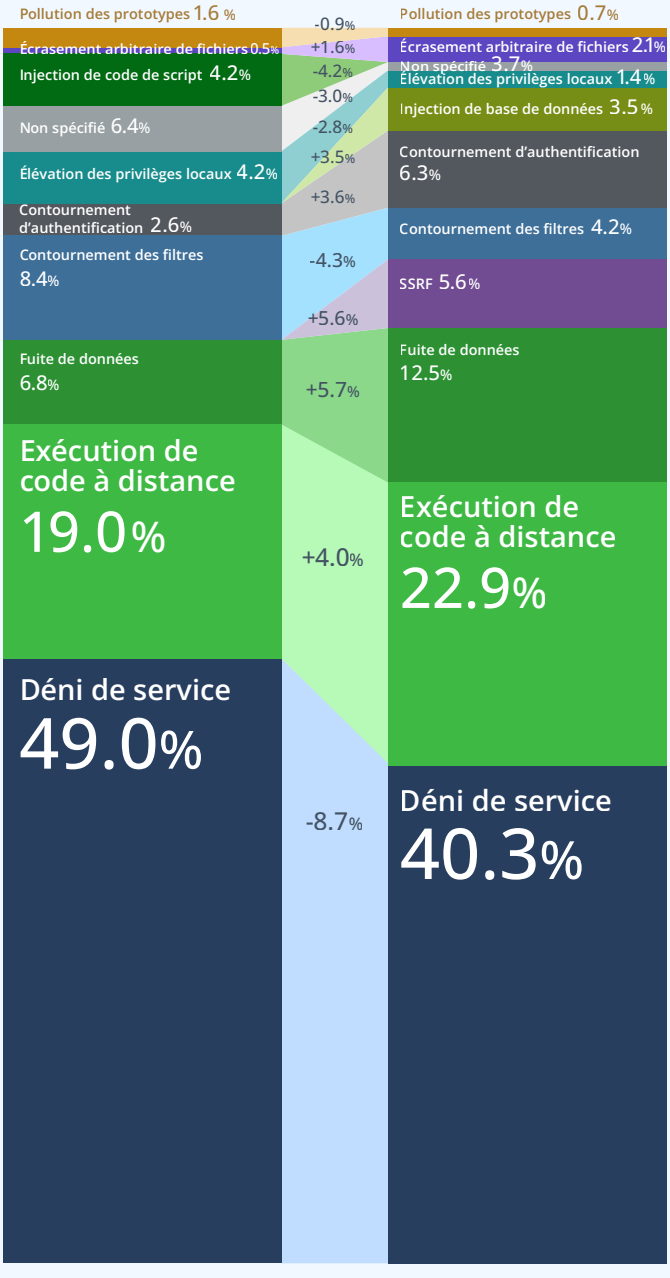


Figure 9. Impacts des vulnérabilités communes pour les CVE les plus connues en 2023 et 2024

Cette année, l'équipe de recherche en sécurité de JFrog a analysé un peu plus de 140 CVE à haut profil (HPCVE), en se basant sur leur pertinence et leur impact potentiel pour nos clients. Le risque de déni de service demeure le principal impact potentiel recensé en cas d'exposition (58). L'exécution de code à distance (33) reste le deuxième impact le plus fréquent, et est passée de 18,9 % à 22,9 %. Il est préoccupant de constater que l'exécution de code à distance représente une part croissante des HPCVE, compte tenu du contrôle potentiellement dévastateur qu'elle peut conférer aux attaquants.

Les fuites de données (18) sont restées en troisième position, mais ont également augmenté en pourcentage total. Le contournement de l'authentification et les attaques SSRF ont eux aussi augmenté, ces attaques SSRF n'ayant été détectées que dans les recherches menées cette année. D'autre part, nous avons constaté une diminution des vulnérabilités liées au contournement des filtres.

L'équipe de recherche en sécurité de JFrog tient compte de plusieurs facteurs lorsqu'elle établit l'ordre de priorité des CVE pour la recherche. L'équipe se concentre sur les technologies pertinentes pour les clients de JFrog et donne la priorité aux problèmes de gravité « à haute risque » et « critique » (c'est-à-dire le score CVSS >= 7,5), mais utilise également une prédiction de la gravité basée sur l'apprentissage automatique lorsqu'un score CVSS n'est pas disponible. L'équipe accorde également la priorité à toute vulnérabilité exploitée activement ou largement médiatisée, même si sa gravité publique est évaluée comme moyenne ou faible.

Gravité des vulnérabilités introduites dans votre chaîne d'approvisionnement logicielle

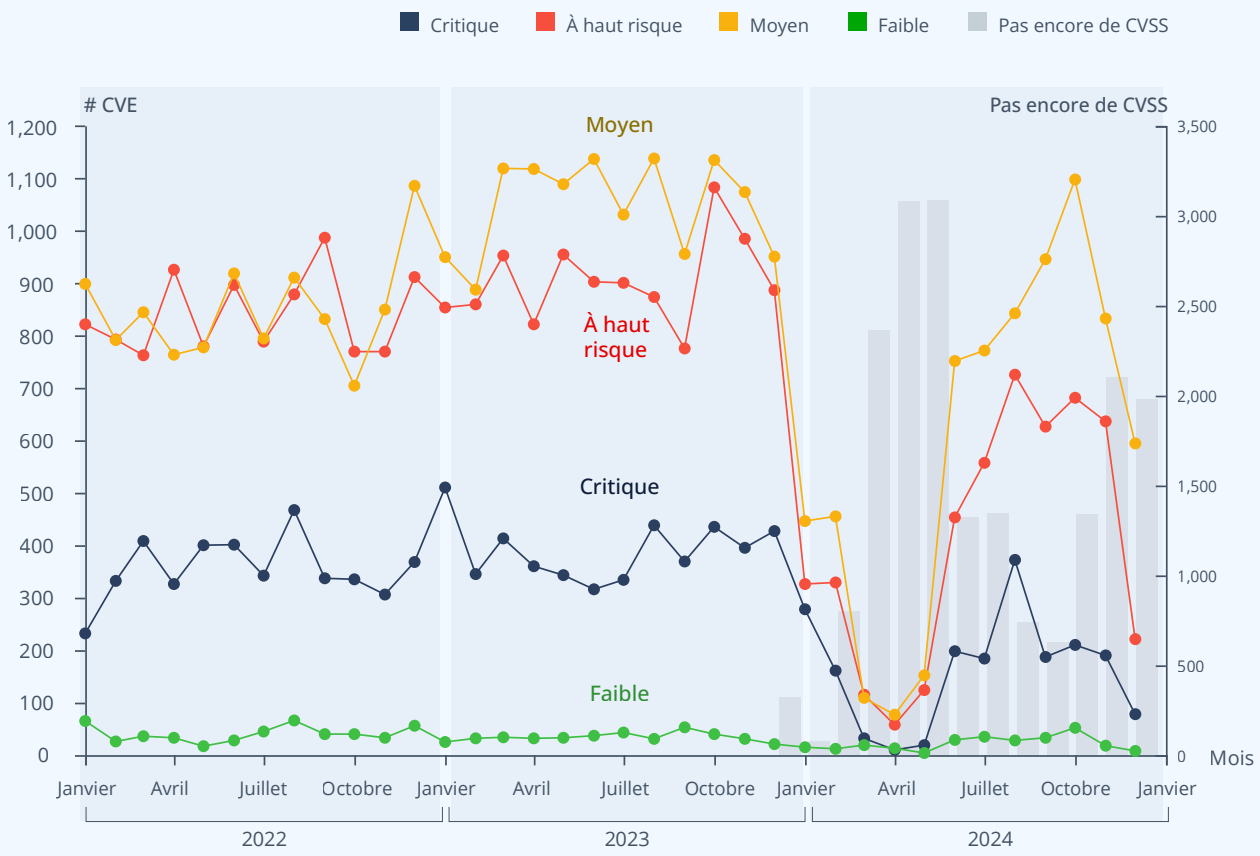


Figure 10.1. CVE par mois et par gravité au cours des 3 dernières années (NVD)

On observe dans les données une forte diminution, puis une remontée des évaluations CVSS à la mi-2024. Toutefois, cela est trompeur. Cette baisse apparente est due au fait qu'en février 2024, la NVD a cessé d'effectuer des recherches sur les CVE tout en se restructurant en raison de réductions budgétaires. Pour résoudre ce problème, la NVD a annoncé en juin 2024 que la CISA avait été engagée pour l'aider dans ses recherches. Si cette perturbation n'avait pas eu lieu, nous aurions certainement vu des chiffres CVSS plus prévisibles.

Pour pallier la pénurie de ressources, la NVD confie désormais la plupart du temps l'évaluation CVSS à des fournisseurs externes connus sous le nom de Authorized Data Publishers (ADPs). Actuellement, la CISA est le premier et le seul éditeur de données. L'équipe de recherche en sécurité de JFrog analyse en permanence les modèles de notation des CVE notés par la CISA, et les premières analyses indiquent que **la CISA donne des notes encore plus exagérées (avec une pondération plus élevée en termes de gravité) que la NVD.** Pour l'avenir, on s'inquiétera également des incohérences potentielles dans la notation CVE à mesure que d'autres fournisseurs de données autorisés apparaîtront. C'est une réalité à laquelle les organisations devront faire face lorsqu'elles détermineront comment prioriser les initiatives de sécurité.

D'après les données NVD disponibles, la tendance reste la même : un nombre élevé de CVE de gravité moyenne et élevée, un nombre faible de CVE de faible gravité, et les CVE critiques se situent à un niveau intermédiaire entre les totaux de gravité faible et moyenne/élevée.

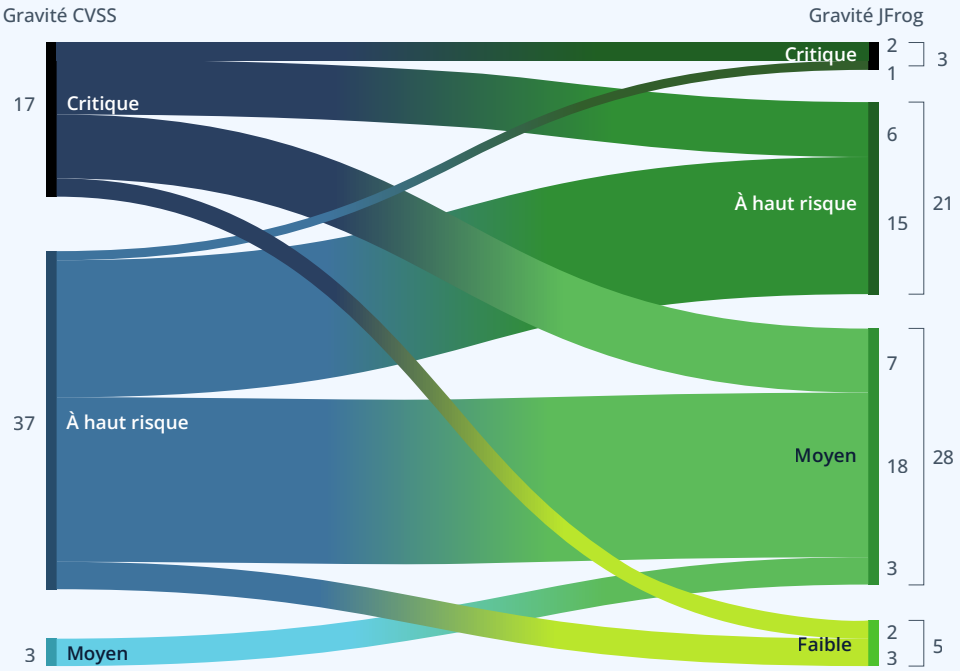


Figure 10.2. Scores de sévérité des CVE (sévérité propriétaire selon JFrog Security Research, comparée à celle de la NVD)

Toutefois, toutes les notes attribuées aux CVE ne reflètent pas toujours la réalité. L'équipe de recherche en sécurité de JFrog évalue régulièrement les CVE pour déterminer leur impact réel et leur attribuer une note de gravité JFrog. La note de gravité JFrog, créé par les experts DevSecOps de JFrog, tient compte des conditions de configuration requises pour que les vulnérabilités soient exploitables.

Les notes CVSS ne tiennent compte que de la gravité d'une exploitation réussie de la vulnérabilité, et non de l'exploitabilité de la vulnérabilité. Parfois, la configuration ou la méthode d'exploitation est un paramètre non standard pour un package ou une dépendance, ce qui rend très improbable l'exploitation de la vulnérabilité. Cette surpondération de la notation CVE reste préoccupante d'une année sur l'autre.

La raison pour laquelle cette tendance à attribuer une note plus élevée aux CVE est préoccupante est qu'aucune explication de la méthode de notation n'a changé. Étant donné que les mécanismes de notation sont essentiels pour déterminer la perception initiale du risque associé à un package, la surpondération des CVE augmente le risque de faux positifs.

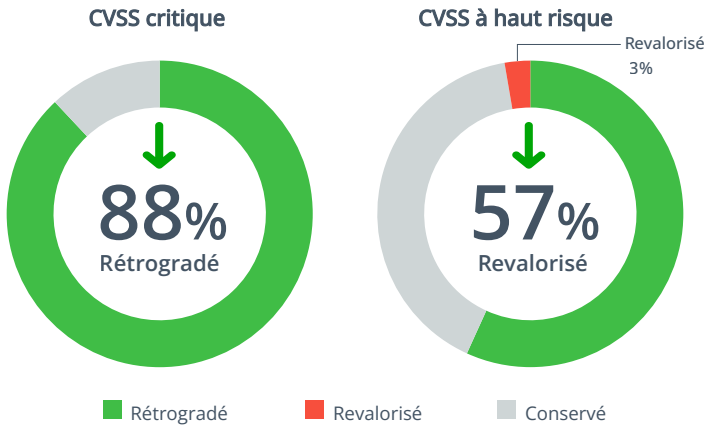


Figure 10.3

Sur la base d'un échantillon de 140 CVE de premier plan, JFrog Security Research a révélé que 88 % des scores CVE critiques et 57 % des scores CVE élevés n'étaient pas aussi sévères que la notation CVSS le laissait supposer.

Taux d'applicabilité des CVE les plus connues

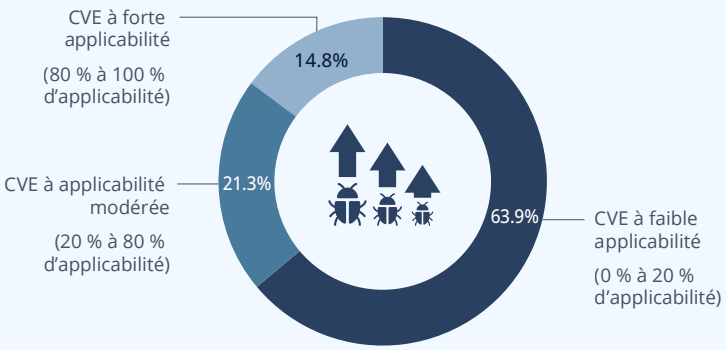


Figure 10.4. Évaluation de l'applicabilité de 183 CVE majeures (méthodologie propriétaire de JFrog Security Research utilisant les bases de données CVE et JFrog)

La simple attribution d'un score de gravité aux CVE ne suffit pas à évaluer l'impact d'une vulnérabilité sur un produit logiciel spécifique. L'équipe de recherche en sécurité de JFrog ne se contente pas d'attribuer des scores de gravité ; elle évalue également les conditions qui influent sur l'exploitabilité de ces vulnérabilités. À cette fin, JFrog crée des scanners d'« applicabilité » qui déterminent si les critères d'exploitabilité sont remplis dans un produit logiciel particulier.

L'équipe JFrog Security Research a développé des scanners d'applicabilité pour 183 CVE rendues publiques en 2024 (CVE-2024-*), en se concentrant sur les CVE à haute risque et critiques concernant les composants et technologies les plus utilisés par nos clients. Ce graphique détaille la fréquence à laquelle la CVE a été jugée applicable (c.-à-d. susceptible d'être exploitée par un acteur malveillant) parmi les clients de JFrog par rapport à non applicable (c.-à-d. non exploitable). Seuls 27 CVE (15 %) se sont révélées hautement exploitables, avec un taux d'applicabilité supérieur à 80 % dans les artefacts analysés par JFrog Xray en 2024. En revanche, 117 CVE (64 %) ont été trouvées avec un faible taux d'exploitabilité et un taux d'applicabilité de 0 % à 20 %.

CVE-2024-24792 est l'un des exemples notables avec un taux d'applicabilité très élevé (99,6 %). Cette vulnérabilité peut être exploitée lors de l'utilisation classique du package de parsing TIFF dans le langage Go et apparaît fréquemment dans les applications qui gèrent le téléchargement et le traitement d'images. Elle est applicable dans la plupart des scénarios, car lorsque la bibliothèque est utilisée pour accepter des images TIFF susceptibles d'être manipulées par un utilisateur, cette CVE peut être déclenchée, entraînant une panique dans l'application.

En revanche, CVE-2024-45490 (lié à Expat, un analyseur XML écrit en C) est l'une des CVE les moins applicables, avec moins de 10 % des cas jugés applicables. L'exploitation de cette vulnérabilité par un attaquant nécessite la modification du paramètre « len » passé à la fonction XML_ParseBuffer() de l'API de la bibliothèque. Cependant, ce scénario est très improbable, car les développeurs fournissent généralement la longueur du document XML eux-mêmes, souvent en utilisant des fonctions comme « stat » ou « XML_GetBuffer ».

Types d’applicabilité des CVE de premier plan

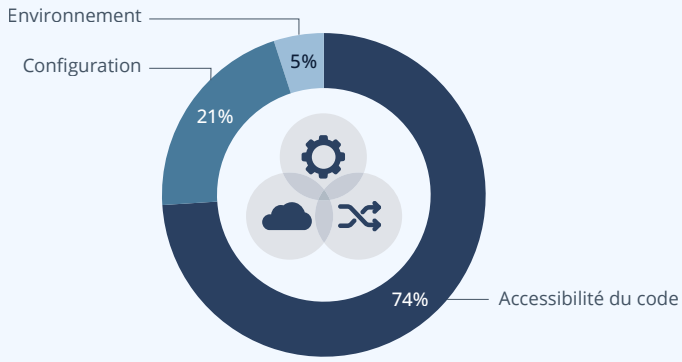


Figure 10.5. Types d’applicabilité des CVE en 2024 (méthodologie propriétaire de JFrog Security Research utilisant les bases de données CVE et JFrog)

L’équipe de recherche en sécurité de JFrog a également étudié la manière dont ces CVE seraient accessibles et exploitables dans une application. Déterminer si une vulnérabilité est applicable ou exploitable nécessite plus qu’une simple analyse de la portée d’accès au code vulnérable via les méthodes traditionnelles d’analyse de l’accessibilité des appels. Il est essentiel d’examiner également les paramètres de configuration des applications et des bibliothèques, ainsi que les conditions environnementales du système d’exploitation sous-jacent. Cette approche holistique garantit une évaluation complète des risques potentiels, alors que la simple identification du code accessible néglige des facteurs critiques qui peuvent influencer de manière significative l’exploitation des vulnérabilités.

Par exemple, dans le fameux « Sudoedit bypass », CVE-2023-22809, l’applicabilité de la vulnérabilité ne peut être déterminée qu’en examinant le fichier de configuration de Sudo (« sudoers ») et en recherchant une configuration spécifique qui n’est pas celle par défaut. Il n’y a aucun moyen de déterminer si la vulnérabilité est applicable en examinant l’accessibilité du code, car le composant vulnérable « sudo » est un utilitaire autonome et non une bibliothèque de code qui peut être invoquée par un code tiers.

Certains packages malveillants sont plus dangereux que d'autres

Dans notre rapport 2024, nous avions souligné la prévalence des packages malveillants dans l'écosystème npm. Un bilan de fin d'année 2024 sur les principaux écosystèmes de packages confirme que npm reste le pire élève en ce qui concerne la présence de packages malveillants. Il convient de mentionner, ce qui n'est peut-être pas surprenant compte tenu de l'augmentation rapide de sa popularité, que le nombre de modèles malveillants chargés sur l'écosystème Hugging Face a augmenté d'environ 6,5 fois au cours de l'année. Voici trois attaques malveillantes remarquables qui ont mérité l'attention de l'équipe de recherche en sécurité de JFrog :

Porte dérobée de XZ Utils

Le 29 mars, une importante faille de sécurité a été signalée dans XZ Utils, un package largement utilisé dans les principales distributions Linux, qui contenait un code malveillant permettant un accès SSH à distance non autorisé. La porte dérobée sophistiquée, présente dans les versions 5.6.0 et 5.6.1, modifiait les routines du serveur OpenSSH pour permettre à certains attaquants d'exécuter des charges arbitraires avant l'authentification, prenant ainsi le contrôle effectif des machines victimes.

Source >

Docker Hub

De récentes campagnes de malwares ciblant Docker Hub ont conduit à la création de millions de dépôts « sans image », contenant des métadonnées malveillantes à la place des images de conteneur. Il est alarmant de constater que près de 20 % (environ trois millions) de ces dépôts publics hébergeaient des contenus nuisibles, allant du spam promouvant du matériel piraté aux logiciels malveillants et aux sites d'hameçonnage, chargés par des comptes automatisés.

Source >

Hugging Face

La surveillance des modèles d'IA a permis d'identifier une famille de modèles qui exécute du code lors du chargement d'un fichier Pickle, offrant ainsi aux attaquants un shell de retour et un contrôle total de la machine compromise via une porte dérobée. Cette infiltration silencieuse présente des risques importants, car elle permet potentiellement d'accéder à des systèmes critiques, ce qui conduit à des violations de données à grande échelle ou à l'espionnage d'entreprise, tout en laissant les victimes dans l'ignorance de la compromission.

Source >

Autres sources de risque cachées dans votre code

Les équipes CISO et AppSec savent déjà qu'il est important d'examiner minutieusement ce que l'on importe de la communauté open source. Cependant, ce n'est pas le seul domaine auquel il faut prêter attention pour assurer une sécurité applicative globale.

Mauvaises configurations et erreurs – l'impact de l'erreur humaine

2024 a connu son lot d'incidents de sécurité au cours desquels des données ont été exposées en raison de fuites de données, d'expositions et de mauvaises configurations.

Avril 2024

Home Depot a été victime d'une violation de données après qu'un fournisseur SaaS tiers a divulgué un sous-ensemble de données relatives aux employés, exposant ainsi les informations personnelles de 10 000 employés.

Source >

Septembre 2024

Des données appartenant à env. 2000 clients de Fortinet stockées sur un site Azure SharePoint ont été consultées par un hacker qui les a ensuite divulguées sur Internet.

Source >

Août 2024

Des milliers de clients d'Oracle NetSuite ont divulgué par inadvertance des données sensibles à des utilisateurs non authentifiés par le biais de boutiques externes construites avec NetSuite SuiteCommerce ou NetSuite Site Builder.

Source >

Septembre 2024

Une exposition importante de données affectant potentiellement des millions d'utilisateurs a été découverte au sein de Microsoft Power Pages, une plateforme SaaS à code bas, en raison de contrôles d'accès mal configurés.

Source >

Septembre 2024

Plus de 1 000 instances ServiceNow mal configurées ont été découvertes, exposant à des utilisateurs externes et à d'éventuels acteurs malveillants des articles de la base de connaissances contenant des informations sensibles sur l'entreprise.

Source >

Décembre 2024

La fuite de données impliquant l'entreprise de logiciels automobiles de Volkswagen, Cariad, se distingue comme l'une des erreurs de configuration SaaS les plus impactantes de 2024. Cet incident a exposé les données collectées sur environ 800 000 voitures électriques, y compris la localisation précise des véhicules et des informations pouvant être liées aux noms des conducteurs.

Source >

État des fuites de secrets dans les artefacts binaires

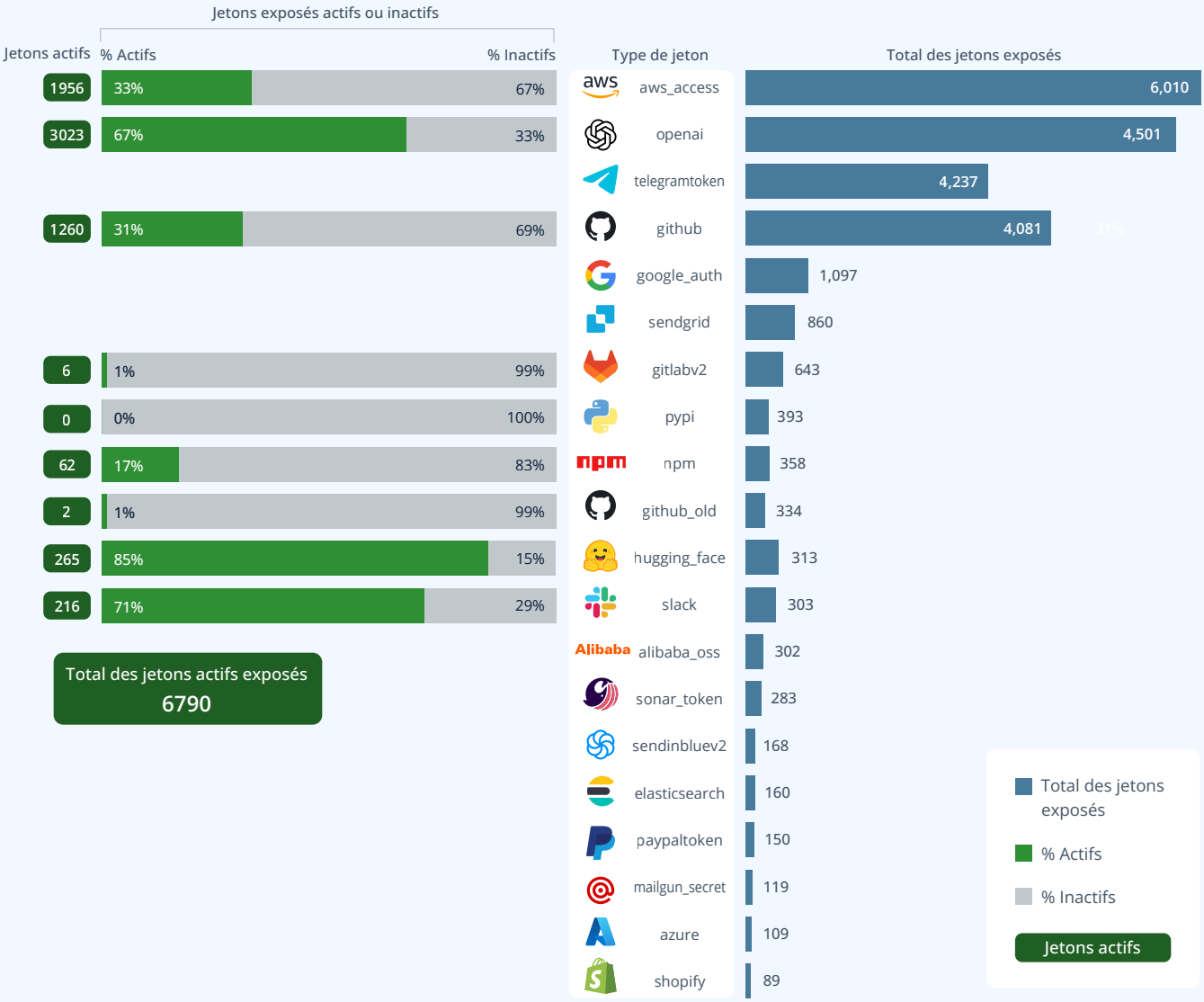


Figure 11.1. Les 20 types de jetons les plus exposés en 2024

L'équipe de recherche en sécurité de JFrog a analysé des millions d'artefacts dans les registres de logiciels open source les plus courants : DockerHub, npm et PyPI. Cette année, les membres de l'équipe ont également noté où se trouvaient les jetons actifs (c'est-à-dire les jetons qui pouvaient être utilisés au moment de la collecte des données).

En comparaison avec l'an dernier, tous les types de jetons découverts sont en hausse ou presque, avec une augmentation de 66 % du total de secrets exposés d'une année à l'autre.

Les jetons les plus exposés sont les mêmes que dans notre dernier rapport, et ont également connu des augmentations significatives d'une année à l'autre : AWS (augmentation de 70 %), OpenAI (augmentation de 103 %), Telegram (augmentation de 62 %) et GitHub (augmentation de 82 %). Les jetons GCP ont également connu un pic important, en hausse de 86 % par rapport à l'année précédente.

Les jetons Hugging Face sont un nouveau type de jeton qui a été ajouté aux scanners de l'équipe de recherche en sécurité de JFrog cette année, ce qui témoigne de la popularité croissante des modèles et des ensembles de données open source. Les jetons Hugging Face représentent le pourcentage le plus élevé de jetons actifs par rapport aux autres jetons de la liste (~85 % d'actifs). L'équipe de recherche en sécurité de JFrog a notamment identifié 6 790 secrets actifs au moment de la collecte des données, révélant ainsi une énorme source potentielle d'accès à des systèmes propriétaires pour des acteurs malveillants.

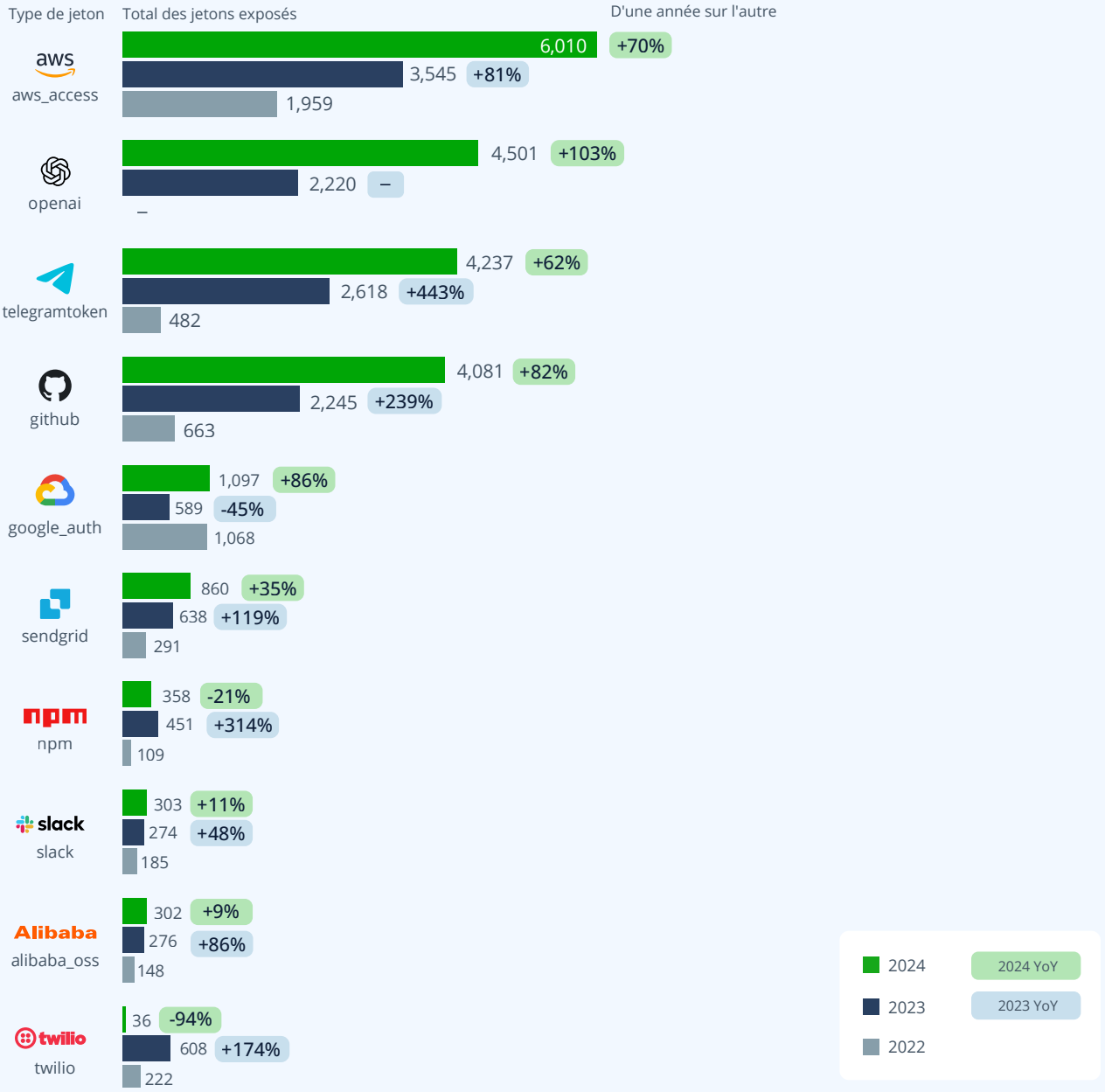
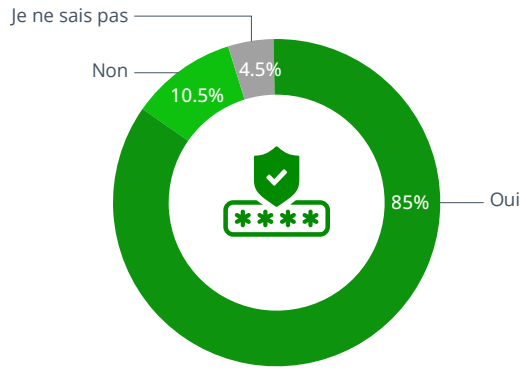


Figure 11.2. Comparatif d'une année sur l'autre des jetons les plus fréquemment exposés

Votre organisation a-t-elle mis en place des mesures de sécurité pour détecter les secrets laissés dans les bases de code et/ou les fuites de jetons ? (Enquête commanditée, 2024)

Les organisations investissent spécifiquement pour que les secrets ne tombent pas entre les mains d'acteurs malveillants ou du public. Toutefois, 15 % des organisations n'ont pas mis en place de mesures ou n'en sont pas sûres. Compte tenu de l'état des secrets divulgués et du fait que nous avons constaté des augmentations pour presque tous les types de jetons découverts, il s'agit d'une part importante des répondants qui se rend vulnérable.



Quelle est la gravité d'une fuite de secrets ?

En juin 2024, l'équipe de recherche en sécurité de JFrog a découvert et signalé une fuite de jeton d'accès avec un accès administrateur aux dépôts GitHub de Python, PyPI et Python Software Foundation, qui a été divulguée dans un conteneur Docker public hébergé sur Docker Hub.

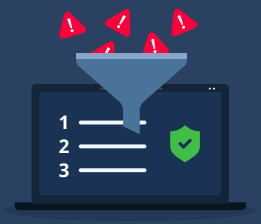
Dans le cadre de son engagement envers la communauté, l'équipe de recherche en sécurité JFrog analyse en continu des dépôts publics tels que Docker Hub, npm et PyPI afin d'identifier des packages malveillants et des secrets divulgués. L'équipe signale toute découverte aux responsables concernés avant que les attaquants ne puissent en tirer parti. Bien que JFrog découvre fréquemment des secrets divulgués de manière similaire, ce cas était exceptionnel : il est difficile de surestimer les conséquences potentielles si ce secret était tombé entre de mauvaises mains ; il aurait, en théorie, été possible d'injecter du code malveillant dans tous les packages PyPI, voire dans le langage Python lui-même.

L'équipe de recherche en sécurité de JFrog a identifié la fuite et l'a rapidement signalée à l'équipe de sécurité de PyPI, qui a révoqué le jeton en l'espace de 17 minutes seulement. Bien que la catastrophe ait été évitée cette fois-ci, l'incident rappelle brutalement le potentiel catastrophique qu'une simple fuite de secret peut déclencher. PyPI étant l'un des principaux dépôts au monde, les répercussions auraient pu être considérables, affectant d'innombrables utilisateurs et projets. Si une telle violation peut se produire dans Python/PyPI, qui est une infrastructure hautement entretenue et largement utilisée, cela souligne la vulnérabilité qui existe sur toutes les plateformes et dans tous les langages, et met en évidence le fait que cette menace peut frapper n'importe qui à n'importe quel moment.

Éléments clés

Le besoin de redondance des données

Les organisations et les outils de sécurité qui s'appuient uniquement sur les données de vulnérabilité de la NVD risquent de manquer des informations CVE essentielles en raison des retards importants enregistrés par la NVD au cours de l'année écoulée. Ces retards signifient que les vulnérabilités nouvellement divulguées et leurs impacts potentiels peuvent ne pas être rapidement inclus dans la base de données, laissant les organisations exposées sans le savoir à des menaces émergentes. Pour atténuer ce risque, il est essentiel de compléter les données de la NVD par d'autres sources, telles que les avis des fournisseurs et les flux de renseignements sur les menaces, afin d'être informé en temps utile des vulnérabilités critiques. Les organisations doivent évaluer les sources de données de tout outil de scan de sécurité afin de garantir une couverture étendue et une redondance optimale.



Applicabilité, impact, hiérarchisation

Avec un volume toujours croissant de CVE à gérer, les équipes de sécurité et de développement risquent d'être paralysées en essayant de trier chaque vulnérabilité. Il est crucial de comprendre l'applicabilité, le vecteur d'attaque et l'impact potentiel d'une CVE dans une application pour concentrer les efforts sur la correction des vulnérabilités qui comptent réellement. L'équipe de recherche en sécurité JFrog continue de constater une surestimation des risques dans les scores CVSS, une tendance qui semble s'accélérer depuis que la CISA contribue désormais à enrichir les enregistrements CVE en tant que premiers éditeurs de données autorisés.

La divulgation de secrets peut arriver à tout le monde

Les organisations doivent rester vigilantes en matière de protection contre les secrets exposés et s'efforcer d'étendre les protections aux développeurs travaillant sur des projets personnels et communautaires. Même si le système d'un développeur est compromis dans le cadre d'un projet personnel, l'impact peut s'étendre aux systèmes de l'entreprise. Les conséquences de la découverte d'un secret exposé ou d'un jeton divulgué peuvent être extrêmement graves. Dans le cas du secret Python/PyPI découvert par JFrog, le détenteur d'un tel jeton aurait eu un accès administrateur à tous les dépôts de Python, PyPI et Python Software Foundation, ce qui aurait pu permettre de mener une attaque à très grande échelle de la chaîne d'approvisionnement logicielle. Si cela peut arriver à Python/PyPI, peut arriver à n'importe qui.

Sophistication des acteurs malveillants

Les acteurs malveillants redoublent de créativité et d'ingéniosité pour infiltrer la chaîne d'approvisionnement logicielle. Dans le cas de la porte dérobée XZ Utils, l'attaquant s'est forgé une réputation crédible de développeur OSS sur plusieurs années et a utilisé un code hautement obscurci afin d'éviter d'être détecté par les revues de code. D'autres acteurs exploitent les outils d'IA en identifiant les cas où les assistants de code IA recommandent des bibliothèques « hallucinées », puis en créant rapidement cette bibliothèque contenant du code malveillant. Les organisations doivent rester vigilantes, même dans le cas de projets open source bien considérés, et définir des politiques de risque opérationnel pour éviter que des bibliothèques « crées du jour au lendemain » ne soient accidentellement intégrées à leur chaîne d'approvisionnement.

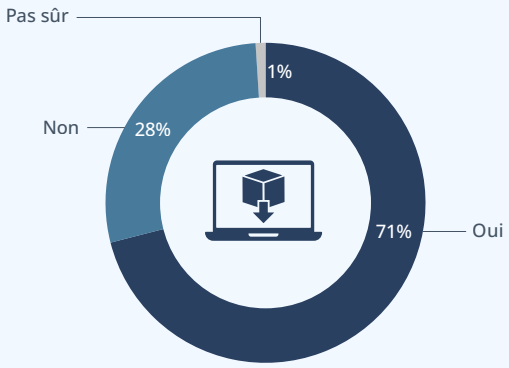
Comment les organisations mettent en œuvre leurs efforts de sécurité aujourd'hui

Cette année, nous avons interrogé 1 402 professionnels de la sécurité, du DevOps et de l'ingénierie. Nous avons élargi les questions de notre enquête et intégré les résultats d'autres rapports de recherche parrainés par JFrog afin d'obtenir une vision plus holistique de la manière dont les équipes gèrent les risques liés aux applications tout au long du cycle de vie du développement logiciel (SDLC). Bien que la plupart des équipes aient mis en place des cadres et des outils de sécurité, nous avons été surpris de découvrir la prévalence de certaines activités à risque telles que le téléchargement de packages ou de bibliothèques de tiers directement à partir d'Internet.

Restrictions en matière d'approvisionnement

L'une des meilleures choses que les entreprises puissent faire pour gérer les risques dans leur chaîne d'approvisionnement logicielle est d'empêcher tout risque d'y pénétrer. Cela va au-delà du « shift left », qui consiste généralement à intégrer les pratiques de sécurité dès la phase de développement, et exige d'aller encore plus loin, en intervenant « en amont du shift left » afin de bloquer les risques avant même qu'ils ne puissent pénétrer la chaîne d'approvisionnement logicielle.

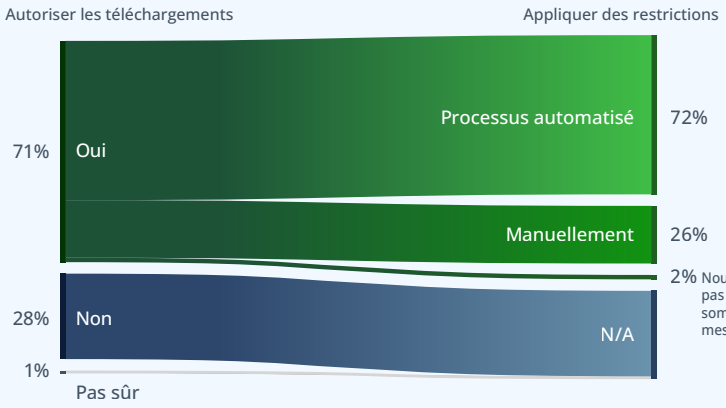
Q **Votre organisation permet-elle aux développeurs de télécharger des packages ou d'autres composants logiciels directement à partir de registres publics ou d'autres sources sur Internet ?**
(Enquête commanditée, 2024)



Il est alarmant de constater que 71 % des entreprises autorisent leurs développeurs à télécharger des composants logiciels à partir d'Internet. La meilleure pratique consiste à empêcher les développeurs de télécharger des packages ou des bibliothèques directement depuis Internet, car le risque est tout simplement trop grand, avec la possibilité d'exposer toute une organisation à des attaques par le biais de la machine d'un seul développeur. La traçabilité est également compromise, car il n'y a aucun moyen de savoir ce qu'un développeur télécharge si vous l'autorisez à le faire directement à partir d'Internet.

Cependant, le fait qu'autant d'organisations l'autorisent indique qu'il existe un réel besoin. C'est là qu'une solution de gestion des artefacts capable de se substituer aux registres publics en amont peut s'avérer particulièrement utile. Un dépôt d'artefacts avec des capacités de proxy agit comme un point de contrôle central pour chaque composant qui entre dans la chaîne d'approvisionnement logicielle, permettant aux équipes de les suivre et de les sécuriser. Avec ce type de solution, les organisations peuvent atténuer en toute sécurité le risque associé à ce type d'activité et prévenir des dommages potentiels d'une ampleur incalculable.

Q **Comment votre organisation assure-t-elle le suivi et l'application des restrictions en matière d'approvisionnement pour les progiciels ou autres composants logiciels provenant directement de registres publics ou d'autres sources sur Internet ?**
(Enquête commanditée, 2024)



Plus d'1 personne interrogée sur 4 (26 %) déclare que son organisation suit et applique manuellement les restrictions d'approvisionnement pour les packages ou autres composants logiciels provenant directement de registres publics ou d'autres sources sur Internet.

Le pourcentage de 72 % des répondants qui indiquent avoir automatisé des processus est étonnamment élevé, mais il peut être influencé par l'étape du SDLC à laquelle le répondant se réfère. Par exemple, il est possible que les développeurs examinent manuellement les packages et les dépendances avant de les intégrer dans leur base de code, mais des processus automatisés peuvent être exécutés pendant le cycle CI/CD ou plus tard pendant les audits des builds de versions. Le problème de cette approche est qu'elle entraîne un travail supplémentaire pour le développeur et qu'elle peut entraîner des risques dès le début du SDLC, comme nous l'avons déjà mentionné dans ce rapport.

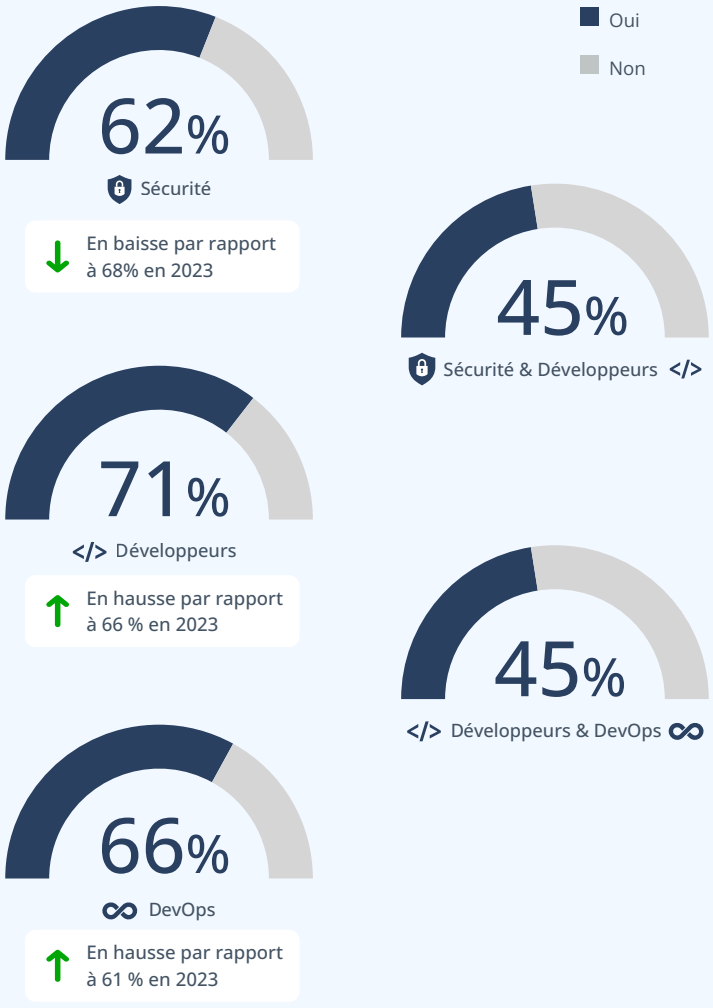
Sur l'ensemble des répondants, 19 % indiquent qu'ils autorisent les développeurs à télécharger des packages directement à partir d'Internet et qu'ils utilisent également des approches **manuelles** pour appliquer les restrictions en matière de sources d'approvisionnement. Cela nécessite un travail manuel important et difficile et ne constitue pas une approche efficace pour bloquer les risques.

Ceux qui déclarent permettre aux développeurs de télécharger des composants directement à partir d'Internet et qui utilisent également des processus **automatisés** pour suivre et appliquer les restrictions en matière d'approvisionnement représentent 52 % des répondants.

Qui gère le processus d'obtention de la dernière version des packages, des bibliothèques et des frameworks : la sécurité ou les développeurs ? Cochez toutes les cases pertinentes. (Enquête commanditée, 2024)

Par rapport aux années précédentes, il semble que les développeurs jouent désormais un rôle plus actif dans la gestion des derniers packages, mais les professionnels de la sécurité conservent leur responsabilité, surtout lorsqu'il s'agit d'examiner et d'approuver les packages à utiliser. Les organisations notent généralement une combinaison d'équipes responsables du processus d'acquisition, avec une répartition égale entre « Développeurs + Sécurité » et « Développeurs + DevOps ».

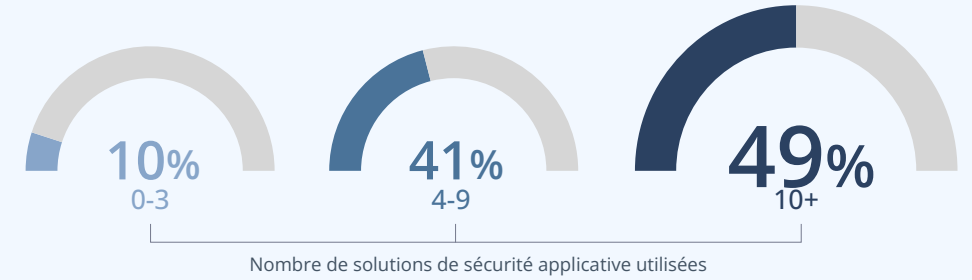
Afin de gagner en rapidité, les entreprises doivent adopter des approches et des solutions qui permettent aux développeurs d'introduire les nouvelles et dernières versions des bibliothèques en open source et d'automatiser l'approbation des packages conformes aux politiques de sécurité. Elles doivent également soutenir un programme de gestion des dérogations qui permette à l'équipe responsable, qu'il s'agisse de l'AppSec ou de la sécurité, d'intégrer ce programme de dérogations dans leur stratégie globale de gestion des risques.

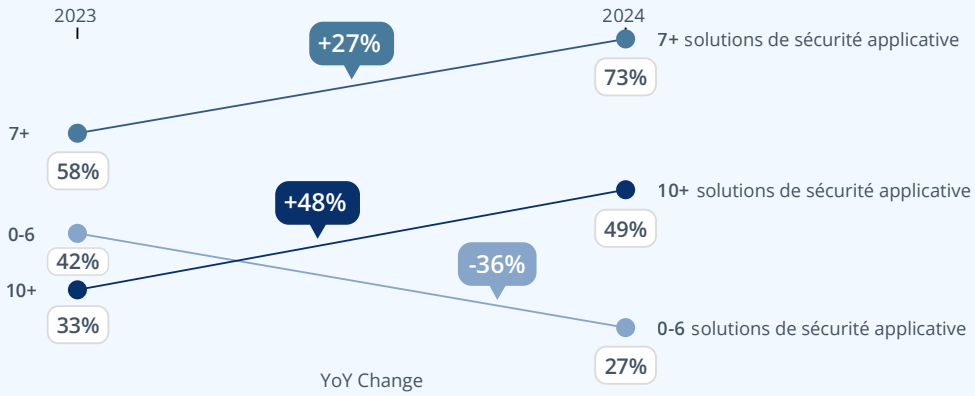


Scan, scan, scan

Bien que les organisations utilisent plus d'outils de sécurité que jamais, il existe toujours des lacunes en matière de couverture. L'absence de scan du code et des binaires, ainsi que l'incohérence de l'analyse dans le cadre du SDLC et de la production, sont des points faibles courants.

Combien de solutions de sécurité applicative utilisez-vous ?
(Enquête commanditée, 2023 et 2024)





Les répondants déclarent utiliser davantage de solutions de sécurité applicative en 2024 par rapport à 2023. À la fin de l'année 2024, 73 % déclarent utiliser 7 solutions de sécurité applicative ou plus, contre 58 % l'année précédente.

Cela va à l'encontre de ce que l'on pourrait attendre compte tenu de l'importance accordée par le marché à la consolidation des outils et de ce que nous disent les responsables des clients de JFrog quant à leur volonté de rationaliser un processus de développement logiciel sécurisé. Les données suggèrent que les ASPM (Application Security Posture Management), une nouvelle catégorie d'outils qui permettent aux organisations de maintenir plusieurs solutions de scan tout en filtrant les résultats en double, sont utilisés par les organisations pour maintenir une surcouverture afin d'éviter le risque de manquer quelque chose. Cependant, les ASPM ne sont qu'un « pansement » pour la prolifération des outils de sécurité et non une solution. Nous ne prévoyons pas que cette croissance du nombre total d'outils de sécurité utilisés se poursuive l'année prochaine, car les entreprises recentrent leurs efforts de consolidation.

Votre organisation applique-t-elle des analyses de sécurité au niveau du code ou au niveau binaire (ou les deux) ? (Enquête commanditée, 2024)

Scan au niveau du code et des binaires

43%

En baisse par rapport à 56 % en 2023

Scan du code uniquement

29%

En hausse par rapport à 27 % en 2023

Scan des binaires uniquement

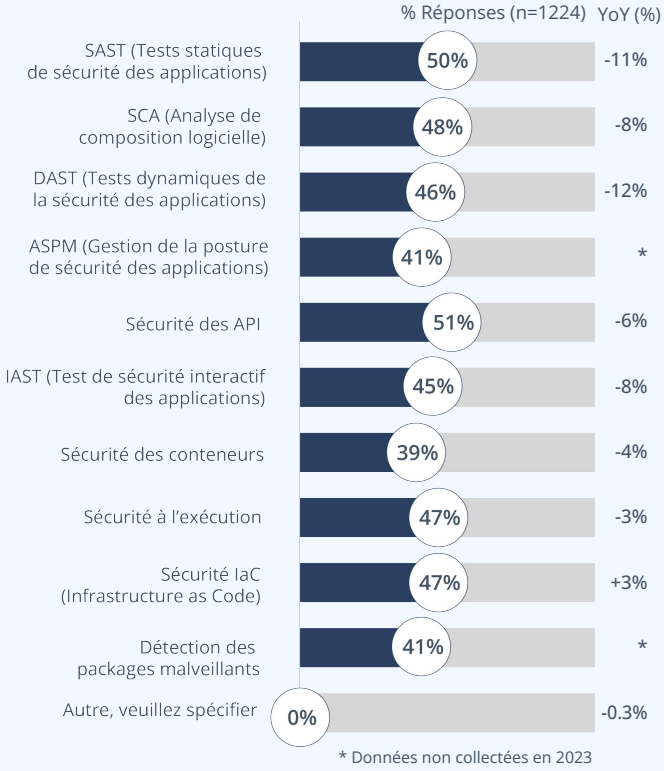
25%

En hausse par rapport à 12 % en 2023

Cette année, l'application d'analyses de sécurité au seul niveau de l'analyse binaire a doublé de popularité : 25 % des personnes interrogées déclarent appliquer la sécurité à ce niveau, contre seulement 12 % en 2023.

43 % des personnes interrogées déclarent que leur organisation applique des analyses de sécurité à la fois au niveau du code et au niveau binaire, ce qui représente une légère baisse par rapport aux 56 % de 2023. Il s'agit d'une tendance quelque peu alarmante, car les organisations devraient idéalement analyser à la fois le code et les binaires afin de prévenir et d'identifier les risques le plus tôt possible. Cette approche se justifie notamment par le fait que certains types de vulnérabilités ne se manifestent qu'au niveau des fichiers binaires. Par exemple, des secrets injectés dans les binaires ou encore des corruptions de mémoire causées par le compilateur, peuvent entraîner des problèmes de sécurité qui ne figurent pas dans le code source ou se retrouver accidentellement dans les builds déployés en production.

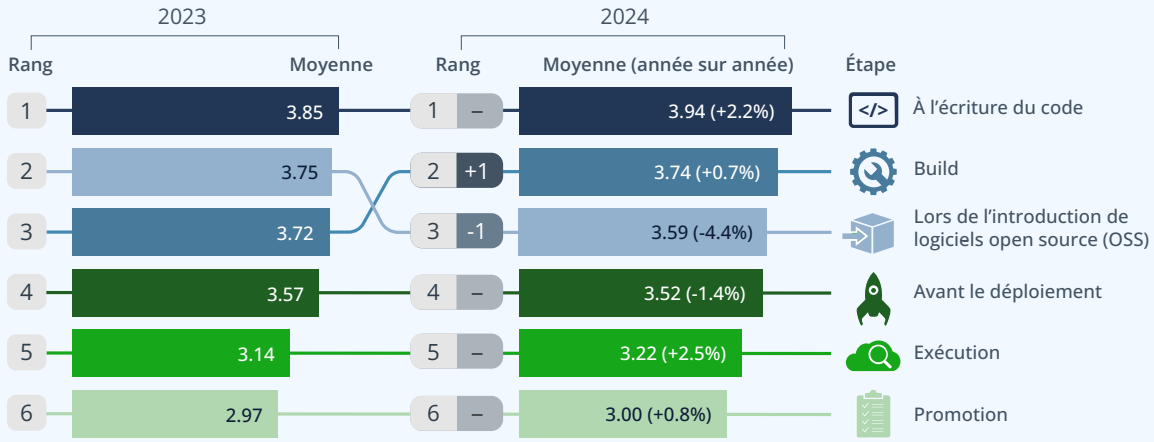
Quels types de solutions de sécurité applicative utilisez-vous ? (Enquête commanditée, 2024)



* Données non collectées en 2023

Aucun outil ne fait l'unanimité. La sécurité API et SAST sont les deux seuls à atteindre ou dépasser les 50 %. L'accent étant toujours mis sur les efforts Shift Left, l'utilisation élevée de SAST est logique. Il n'est pas non plus surprenant de voir les organisations investir dans des outils de sécurité des API, étant donné la prévalence des applications modernes de microservices où les API constituent un point faible potentiel pour l'exploitation par des acteurs malveillants. Les taux d'utilisation des différents types d'outils restent constants d'une année sur l'autre, mais le pourcentage global de répondants qui indiquent utiliser un type d'outil spécifique a diminué. Ce constat est d'autant plus intéressant que le nombre total d'outils de sécurité est également en augmentation, comme nous l'avons vu précédemment dans ce rapport. Cela peut indiquer un chevauchement des types d'outils utilisés, ou que les différentes équipes ont chacune leurs propres outils de sécurité préférés qui finissent par fournir les mêmes fonctionnalités que les outils préférés des autres équipes. Les organisations devraient envisager un audit des outils de sécurité afin d'identifier les chevauchements ou les lacunes en la matière.

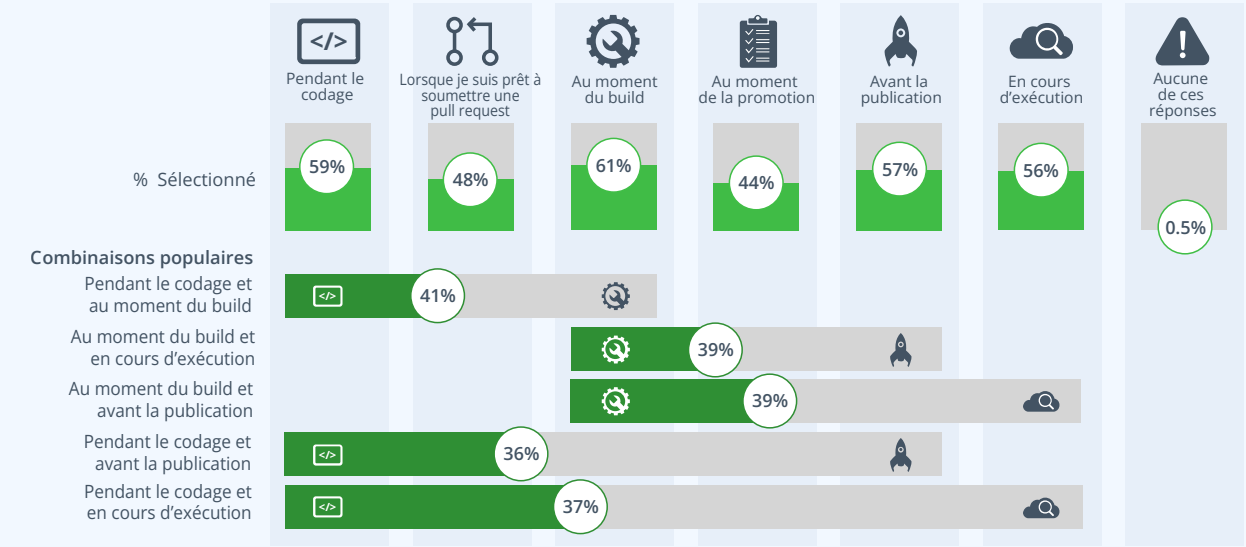
Selon vous, à quelle étape du SDLC faut-il appliquer la sécurité en priorité ? (Enquête commanditée, 2024)



Lorsqu'on leur demande de classer l'étape du cycle de vie du développement logiciel où il est préférable d'appliquer la sécurité, les trois premiers moments signalés restent les mêmes d'une année sur l'autre :

À l'écriture du code	Build	Lors de l'introduction de logiciels open source
M=3.94	M=3.74	M=3.59

À quel stade du développement votre organisation procède-t-elle généralement à des analyses de sécurité ? Cochez toutes les cases pertinentes. (Enquête commanditée, 2024)



"C'est « Pendant le codage » que les organisations procèdent le plus souvent à des analyses de sécurité, près de 3 répondants sur 5 affirmant que leur organisation procède généralement à des analyses de sécurité à ce stade.

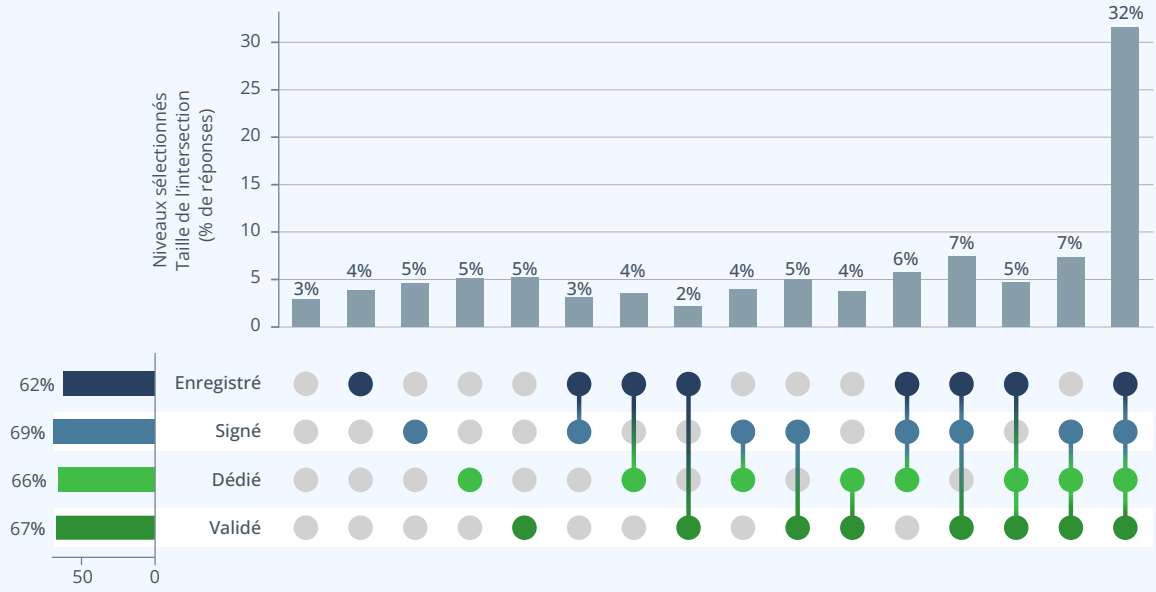
41%	Pendant le codage	+	Au moment du build
39%	Au moment du build	+	Avant la publication
30%	Au moment du build	+	En cours d'exécution
36%	Pendant le codage	+	Avant la publication
37%	Pendant le codage	+	En cours d'exécution

Établir une visibilité et un contrôle sur les pipelines d'applications

Il peut sembler évident que, puisque les entreprises développent des applications, elles doivent gérer les risques de manière globale au niveau applicatif. Cependant, bien que les organisations définissent et suivent déjà les applications tout au long du SDLC, le niveau de contrôle et de traçabilité varie considérablement. Il est essentiel de renforcer ces deux éléments afin de gérer les risques en toute confiance et d'assurer la fiabilité des logiciels diffusés.

Le contrôle commence dès l'étape de l'approvisionnement, avant même qu'une « application » ne voie le jour. Les composants et les bibliothèques qui sont intégrés dans le processus de développement déterminent fondamentalement le niveau de sécurité du produit final. En évaluant et en sélectionnant soigneusement les ressources tierces, les organisations peuvent atténuer les risques bien avant qu'ils ne se matérialisent dans l'application.

Parmi les référentiels de sécurité suivants, lesquels sont mis en œuvre dans votre organisation ? (Enquête commanditée, 2024)

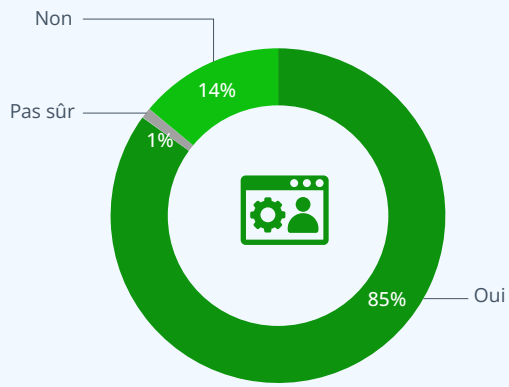


- Enregistré – Les signatures des packages sont validées avant la publication
- Signé – Les packages sont buildés sur un hôte dédié
- Dédié – Les données de build des packages et les métadonnées sont signées
- Validé – Les métadonnées de build des packages sont enregistrées

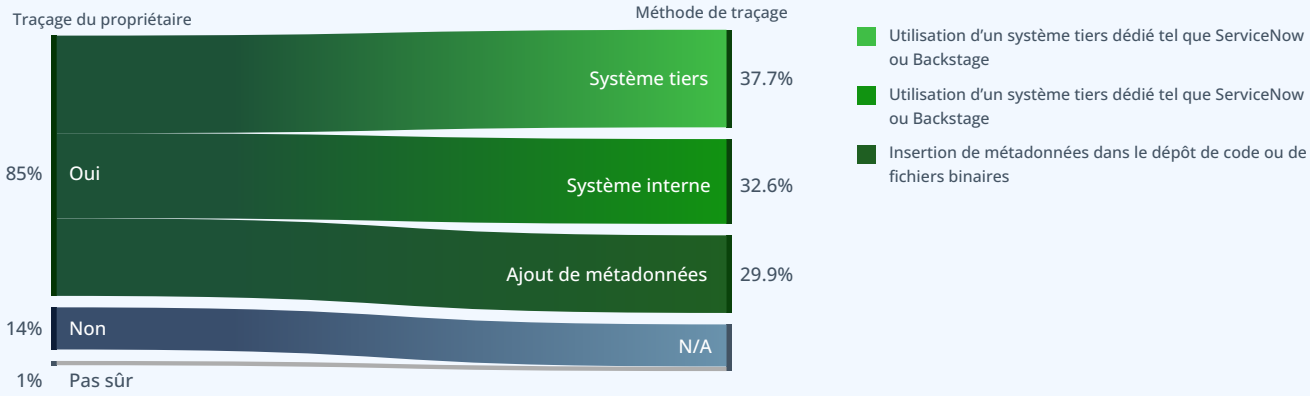
La majorité des organisations interrogées se sont tournées vers des référentiels tels que Supply-chain Levels for Software Artifacts (SLSA), afin d'améliorer la sécurité et l'intégrité des chaînes d'approvisionnement logicielles. Les données montrent une adoption significative d'au moins un niveau du SLSA, tandis qu'un peu plus d'un tiers des répondants adoptent tous les niveaux de du SLSA.

Suivi du propriétaire de chaque application

Pour chaque application que vous créez dans votre organisation, gardez-vous une trace du propriétaire de l'application (c.-à-d., équipe / individus) ? (Enquête commanditée, 2024)



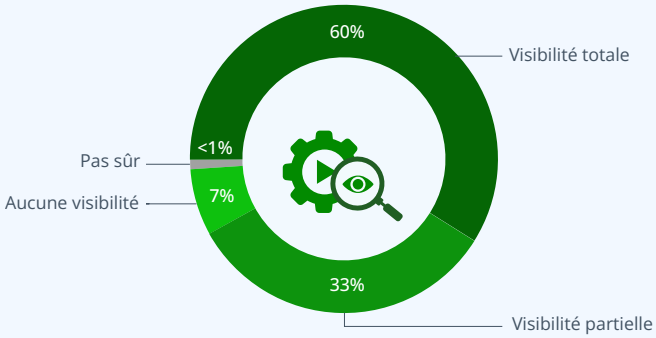
Pour chaque application que vous créez dans votre organisation, comment gardez-vous une trace du propriétaire de l'application ? (Enquête commanditée, 2023 et 2024)



Le traçage des propriétaires d'applications et des différents microservices qui les composent est essentiel pour de nombreuses raisons, notamment pour remédier rapidement aux problèmes, comprendre les interdépendances entre les applications et établir des plans de gouvernance et de continuité des activités appropriés. Si la plupart des organisations tracent le propriétaire des applications qu'elles développent, la manière dont elles le font varie considérablement. Les réponses sont presque également réparties entre les systèmes tiers dédiés, les systèmes internes dédiés et l'utilisation de métadonnées dans le dépôt de code ou de fichiers binaires. Pas un seul répondant ne déclare avoir utilisé une autre méthode que ces trois-là.

Avez-vous une visibilité sur la provenance des logiciels exécutés en production (c'est-à-dire qui a effectué les commits pour un service donné, quels tests et validations ont été réalisés, d'où proviennent les dépendances) ? (Enquête commanditée, 2023 et 2024)

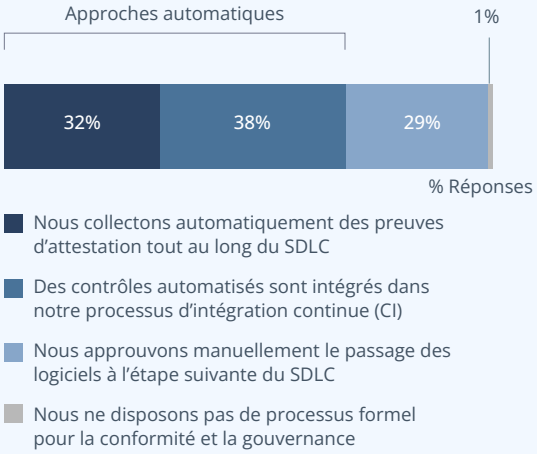
Seules 60 % des organisations déclarent avoir une visibilité totale sur la provenance des logiciels utilisés en production. Environ un tiers (33 %) ont une visibilité partielle et, heureusement, un peu moins de 8 % n'ont aucune visibilité ou ne sont pas sûrs de leur provenance.



La compréhension de la provenance des logiciels est essentielle pour garantir la qualité et la sécurité des logiciels mis sur le marché, et devient rapidement une exigence obligatoire de diverses réglementations gouvernementales. Les quelque 8 % qui déclarent n'avoir aucune visibilité devraient, au minimum, faire l'inventaire de leurs bases de code et de tous les packages externes, et s'assurer qu'ils ont mis en place un système automatisé de CI/CD qui suit et attribue les versions de build. Bien que beaucoup considèrent le contrôle des sources comme une évidence, il est possible qu'une fraction de ces ~8 % n'ait pas encore mis en œuvre une solution robuste de contrôle des sources qui permette de suivre les modifications du code au cours du développement.

Comment garantissez-vous le respect des normes de test et de qualité logicielle tout au long du processus de développement et de déploiement, afin d'assurer la conformité et la gouvernance ? (Enquête commanditée, 2023 et 2024)

Les méthodes utilisées pour garantir les normes de test et de qualité varient également d'une organisation à l'autre. La majorité (70 %) utilise des approches automatisées, mais près d'un tiers (29 %) a encore besoin d'approbations manuelles pour franchir les différentes phases du SDLC.



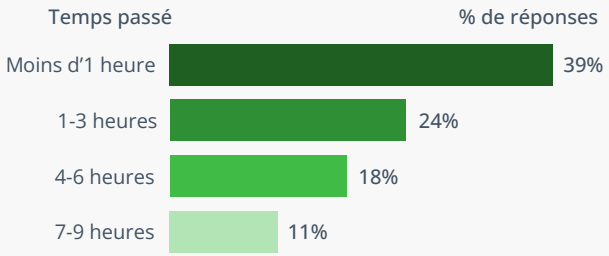
Combien de temps les initiatives de sécurité coûtent-elles à votre organisation ?

Selon une étude menée par IDC et commanditée par JFrog, 60 % des professionnels déclarent que leur équipe de développement et/ou de sécurité consacre généralement 4 jours ou plus à la correction des vulnérabilités des applications au cours d'un mois donné. Cela représente en moyenne un coût de 28 000 dollars par développeur chaque année dédié aux activités de sécurité. Cela n'a pas seulement des conséquences financières, mais aussi un effet négatif sur l'expérience des développeurs (DevEx).

IDC : Le coût caché du DevSecOps
Publication : Septembre 2024 | IDC #US52537524

Temps passé par les développeurs en dehors des heures de travail pour traiter les problèmes de sécurité

Les développeurs passent env. 3,6 heures par semaine en dehors des heures de travail pour traiter les problèmes de sécurité. Cela crée un environnement propice à l'épuisement professionnel.



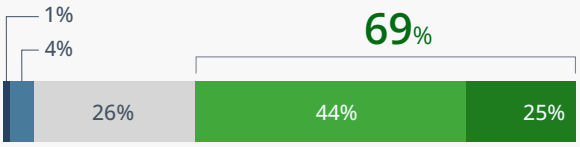
~3.6
heures par semaine passées par un développeur en dehors de ses heures de travail pour traiter les problèmes de sécurité

Argent dépensé pour les activités liées à la sécurité

L'organisation moyenne dépense 28 000 dollars par développeur et par an pour des tâches liées à la sécurité. Bien que le DevSecOps soit un impératif commercial et essentiel pour développer des applications sécurisées, des outils et des processus inefficaces ou mal mis en œuvre font perdre du temps aux développeurs et engendrent des coûts pour l'entreprise.

$28K
dépensés par développeur et par an pour des tâches liées à la sécurité

Trop de changements de contexte



- Tout à fait d'accord
  - Je passe régulièrement d'un outil ou d'un environnement à l'autre
- D'accord
  - Je change souvent d'outil ou d'environnement
- Indécis
- Pas d'accord
  - Je change parfois d'outil ou d'environnement
- Pas du tout d'accord
  - Je quitte rarement ou jamais mes outils ou mes environnements

69 % des développeurs reconnaissent que leurs responsabilités en matière de sécurité les obligent à changer fréquemment de contexte. Alors que les organisations cherchent à améliorer le DevEx, elles devraient considérer que le fait de passer régulièrement d'un outil à l'autre nuira à ces efforts et réduira la probabilité que les développeurs s'engagent dans des activités de sécurité.

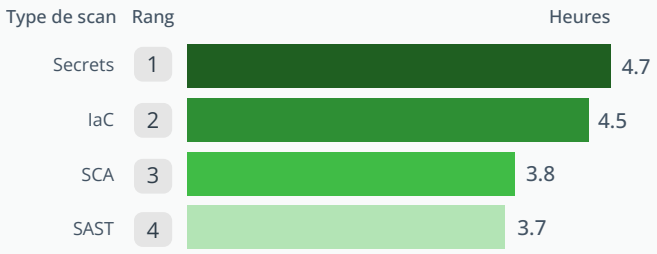
69%
des développeurs reconnaissent que leurs responsabilités en matière de sécurité les obligent à changer fréquemment de contexte

Temps passé par type de scan

Les développeurs consacrent le plus de temps à l'analyse des secrets, ce qui indique soit la nécessité d'une formation plus poussée sur les pratiques de codage pour la gestion des jetons et des secrets, soit la nécessité d'outils de gestion des secrets plus efficaces. Il est également essentiel de s'assurer qu'aucun secret n'est laissé dans le code, ce qui équivaut à déposer les clés de votre maison attachées à un porte-clés indiquant votre adresse. Souvent, les secrets laissés dans le code peuvent donner aux attaquants un accès illimité aux systèmes et aux données critiques.

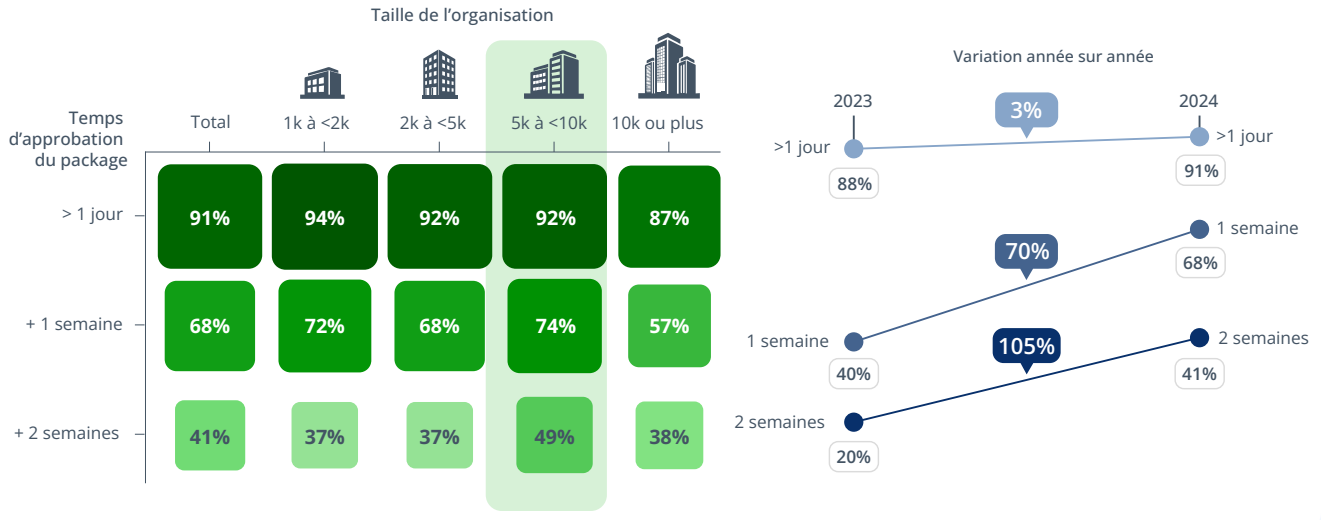
4.7

heures consacrées à l'analyse des secrets par les développeurs



L'étude IDC a été réalisée en ligne en juin 2024 et a recueilli les réponses de 210 développeurs américains et européens, chefs et responsables d'équipes de développement et propriétaires produit axés DevSecOps. L'étude recherchait des informations sur l'impact du temps consacré par les développeurs au DevSecOps, sur les outils et les tâches DevSecOps consommant du temps de développement, sur la valeur du temps consacré par les développeurs au DevSecOps, et sur l'impact des tâches de sécurité sur le flux de travail et la satisfaction des développeurs.

Quel est le délai habituel pour que l'utilisation d'un nouveau package ou d'une nouvelle bibliothèque soit approuvée ? (Enquête commanditée, 2023 et 2024)



Les développeurs attendent plus longtemps que jamais pour l'approbation de nouveau packages. Les entreprises de taille moyenne (de 5K à 10K employés) ont tendance à attendre le plus longtemps, 92 % d'entre elles attendant plus d'un jour, 74 % plus d'une semaine et 49 % deux semaines ou plus. Bien qu'il soit encourageant de voir les développeurs plus impliqués dans le processus, celui-ci reste manifestement inefficace, probablement en raison des révisions et d'autres efforts manuels. Des efforts supplémentaires sont nécessaires pour rendre l'intégration de nouveaux composants réellement en libre-service.

Éléments clés

71%

Pratiques fondamentales manquantes pour la sécurité de la chaîne d'approvisionnement logicielle

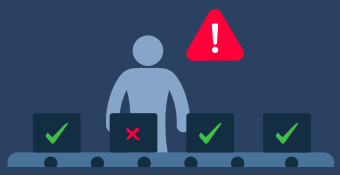
Les entreprises doivent contrôler, ou du moins avoir une bonne visibilité sur ce qui entre dans leur chaîne d'approvisionnement logicielle par l'intermédiaire de leurs développeurs et des dépendances référencées dans leurs applications. Plus de 71 % des organisations autorisent les développeurs à télécharger directement à partir d'Internet, ce qui est préoccupant et constitue une violation majeure des meilleures pratiques en matière de sécurité de la chaîne d'approvisionnement logicielle. Une solution de gestion des artefacts pour les registres publics proxy devrait être mise en place dans chaque organisation.

Plus de scanners, plus de problèmes ?

Les organisations semblent utiliser plus d'outils de sécurité que jamais, mais cela peut-il avoir un impact positif ou négatif sur leur posture de sécurité ? Quoi qu'il en soit, il y a encore des lacunes dans la couverture et de nombreuses organisations n'analysent pas à la fois le code et le niveau binaire, ce qui est problématique.

DevSecOps sans compromettre le DevEx

Les efforts en matière de sécurité prennent chaque semaine des heures de travail aux développeurs. Les organisations peuvent et doivent chercher des moyens de réduire l'impact des efforts de sécurité sur les développeurs sans compromettre la sécurité de leurs applications. Une hiérarchisation intelligente, des résultats contextualisés et l'automatisation sont essentiels dans ce domaine.



Améliorer la gestion des applications

85 % des organisations effectuent un traçage des responsables des applications qu'elles développent en interne, mais les méthodes utilisées pour garantir le respect des standards applicatifs varient considérablement ; près d'un tiers s'appuie encore sur des processus manuels pour faire passer leurs logiciels d'une étape à l'autre. Toute intervention manuelle représente un risque potentiel, qu'il soit intentionnel ou accidentel, et révèle un domaine d'amélioration évident pour les organisations.

La prochaine frontière du risque : le développement de l'IA et du Machine Learning

Presque tous les outils de sécurité et un nombre croissant d'outils de développement vantent désormais les capacités de l'IA pour accélérer le développement et améliorer la détection et la correction des vulnérabilités. Toutefois, dans cette section du rapport, nous nous concentrons sur **la création** d'outils d'IA, plutôt que sur leur utilisation.

La chaîne d'approvisionnement logicielle en intelligence artificielle et en Machine Learning (IA/ML) représente la prochaine frontière du risque pour les organisations, et son niveau de maturité est encore bien inférieur à celui du développement logiciel traditionnel. En effet, selon une étude commanditée par JFrog auprès d'InformationWeek, 79 % des entreprises déclarent que des préoccupations en matière de sécurité freinent l'utilisation et/ou l'intégration de fonctionnalités d'IA/ML dans leurs logiciels.

Tendances en matière d'adoption de l'IA et DevSecOps

L'enquête d'InformationWeek a permis de déterminer dans quelle mesure les développeurs de logiciels et les équipes de cybersécurité comprennent l'importance d'intégrer la sécurité applicative dans le cycle de vie du développement logiciel. Elle s'est également intéressée à la manière dont les équipes protègent leurs organisations contre les codes malveillants et évitent l'utilisation inappropriée des technologies de l'IA. Parmi les enseignements majeurs de ce sondage, on retrouve :

Adoption de l'IA et DevSecOps : Garder une longueur d'avance tout en restant en sécurité

Publication : Septembre 2024 | InformationWeek & JFrog

Manque de confiance dans la sécurité de l'IA au sein de l'entreprise

79% des entreprises déclarent que des préoccupations en matière de sécurité freinent l'utilisation et/ou l'intégration de fonctionnalités d'IA/ML dans leurs logiciels

Les 3 principales préoccupations des entreprises en matière de sécurité concernant l'IA sont l'exposition des données par l'utilisation de LLM, les codes malveillants dans les modèles IA et les biais de l'IA

64% des organisations ne sont pas du tout confiantes ou seulement assez confiantes dans leur capacité à se conformer aux réglementations nouvelles et émergentes concernant l'utilisation de l'IA dans les logiciels

La visibilité de la chaîne d'approvisionnement IA manque de clarté

49% des entreprises n'ont pas de moyen fiable de contrôler l'utilisation des modèles de ML dans leurs applications

Moins de 1/4 des organisations disposent d'une source de vérité unique pour tous les composants logiciels, y compris les modèles d'IA

Plus de 2/3 des organisations n'ont pas de méthode fiable pour effectuer un suivi des packages open source de leurs logiciels contenant des dépendances transitives vers des modèles ML

Insuffisance des politiques concernant l'utilisation l'IA

58% des entreprises n'ont pas de politique en place ou ne savent pas si elles dispose d'une politique fixant des règles pour l'utilisation par les développeurs de modèles ou de composants d'IA open source

60% des entreprises n'ont pas de politique sur la manière dont les développeurs acquièrent ou licencient leurs données d'entraînement

L'application de ces politiques est encore plus aléatoire

68% des personnes interrogées déclarent n'avoir aucun moyen de faire respecter l'utilisation des composants d'IA ou dépendent d'un examen manuel pour ce faire

59% déclarent n'avoir aucun mécanisme ou s'en remettre à un examen manuel pour appliquer les politiques relatives aux données d'entraînement

L'enquête commanditée par JFrog auprès de InformationWeek a été menée en ligne en mai 2024 et a recueilli les réponses de 210 professionnels de l'informatique et de la cybersécurité principalement situés en Amérique du Nord. Les personnes interrogées provenaient d'entreprises de toutes tailles et occupaient des postes allant du cadre supérieur au personnel. Plus de 21 secteurs sont représentés, dont le conseil, la finance, l'éducation, le gouvernement, la technologie, la santé et la production industrielle.

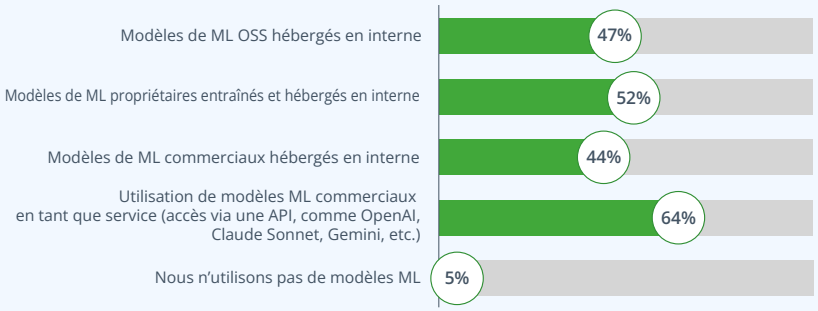
Si l'étude d'InformationWeek a mis en évidence des tendances intéressantes, le reste de cette section se penche un peu plus sur la manière dont les organisations intègrent réellement les services d'IA dans leurs applications et en régissent l'utilisation.

Utilisation, gouvernance et analyse des artefacts de modèles ML

Quelle approche privilégiez-vous pour intégrer des modèles ML dans le cadre du développement de vos applications ? (Enquête commanditée, 2024)

Les méthodes utilisées par les organisations pour mettre en place des services et des applications d'IA varient, et la recherche indique que les organisations utilisent plusieurs méthodes à la fois.

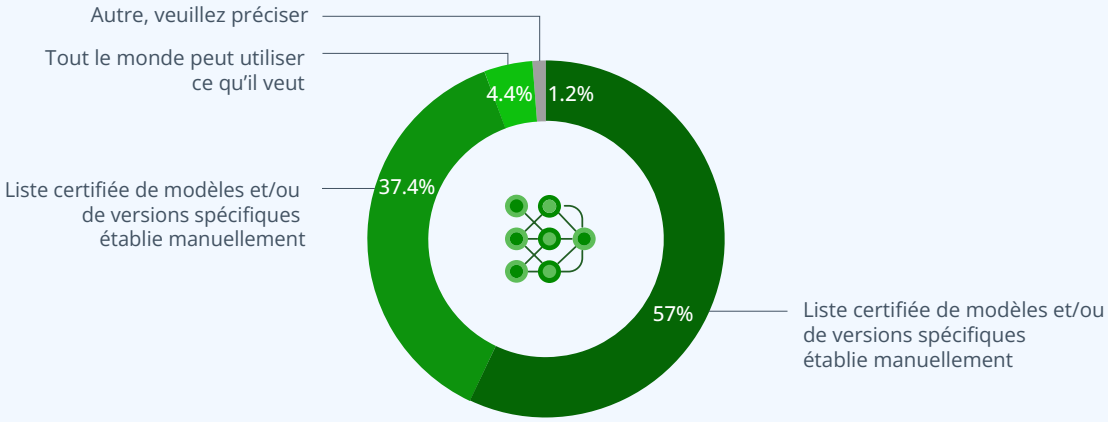
L'approche la plus populaire est de loin l'utilisation de modèles commerciaux accessibles via des API (64 %).



Cela permet aux organisations d'accéder rapidement à des capacités d'IA puissantes et polyvalentes, sans coûts initiaux de développement ou d'infrastructure.

Cependant, nous constatons également que les organisations investissent dans des modèles d'autohébergement, plus de la moitié d'entre elles utilisant des modèles propriétaires conçus pour répondre à leurs besoins spécifiques. Près d'1 personne interrogée sur 2 cite les modèles OSS autohébergés comme leur principale méthode de consommation de modèles de Machine Learning.

De quelle manière encadrez-vous l'usage des artefacts de modèles ML dans votre structure de développement ? (Enquête commanditée, 2024)



Plus d'1 professionnel sur 3 (37 %) déclare régir l'utilisation des artefacts des modèles ML par le biais d'une liste manuelle de modèles et/ou de versions spécifiques.

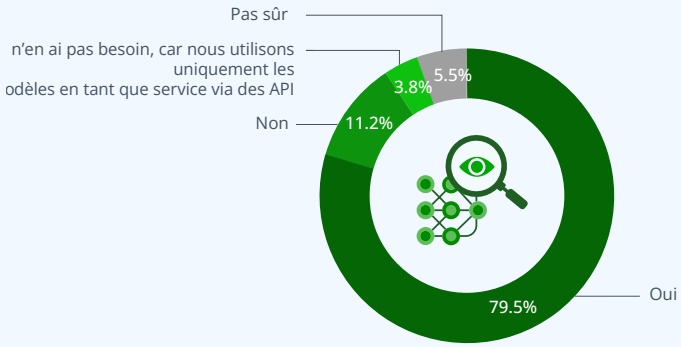
Comme le révèle l'étude d'InformationWeek, 49 % des entreprises n'ont pas de moyen fiable de contrôler l'utilisation des modèles ML dans leurs applications. Cela peut expliquer pourquoi 4 % des personnes interrogées ne prennent volontairement aucune mesure, manuelle ou autre, pour contrôler ce que les développeurs utilisent. Sur l'ensemble de la population interrogée,

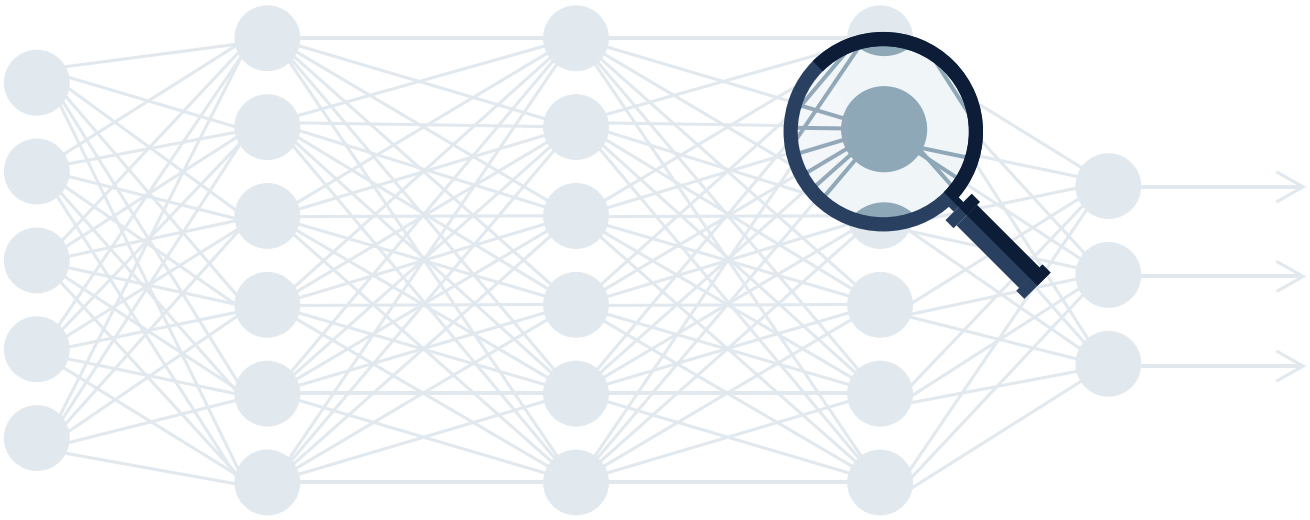
16 % consomment des modèles OSS autohébergés et régissent manuellement l'utilisation des artefacts des modèles.

Votre entreprise a-t-elle mis en place un système de contrôle des artefacts de modèles ML pour identifier d'éventuelles failles de sécurité ou la présence de modèles malveillants ? (Enquête commanditée, 2024)

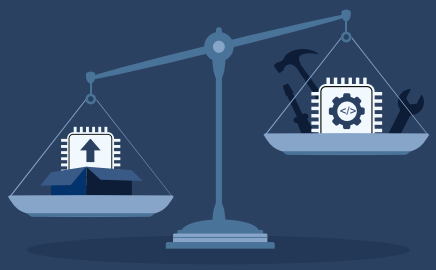
Une grande majorité (79 %) des organisations déclarent avoir mis en place un mécanisme pour analyser les artefacts des modèle ML à la recherche de vulnérabilités de sécurité ou de codes malveillants. 11 % n'ont pas de mécanisme en place.

Heureusement, seuls 3 % des personnes interrogées déclarent consommer des modèles OSS autohébergés et n'avoir mis en place aucune forme de mécanisme de scan pour prévenir les vulnérabilités ou les modèles malveillants. Cependant, il reste encore beaucoup à faire, tant au niveau de l'organisation que de l'industrie de la sécurité, pour s'assurer que les bons outils et les bonnes politiques sont en place pour sécuriser le développement de l'IA/ML.





Éléments clés

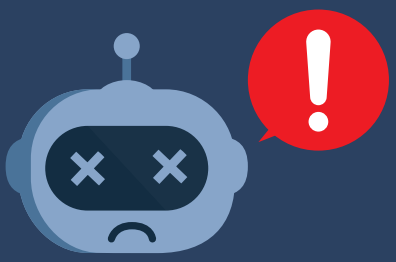


Accélérer l'adoption de l'IA grâce à des modèles commerciaux

Se tourner vers des modèles commerciaux semble être un moyen populaire d'accélérer l'intégration des services d'IA dans les applications commerciales. L'accès aux modèles commerciaux via l'utilisation d'API permet également aux organisations d'économiser du temps et de l'argent lorsqu'il s'agit d'acquérir les outils, les ressources et l'expertise nécessaires pour élaborer et gérer leurs propres modèles internes. En outre, il peut être judicieux pour les organisations qui n'ont pas beaucoup d'expérience en matière d'IA/de ML de confier la sécurité de leurs modèles à des fournisseurs ayant une plus grande expertise dans ce domaine.

La visibilité de la chaîne d'approvisionnement IA manque de clarté

De nombreuses organisations peinent à mettre en place des méthodes fiables pour gérer l'utilisation des modèles d'apprentissage automatique dans leurs applications, et il leur manque souvent une source unique de vérité pour tous les composants logiciels, y compris les modèles ML. En outre, leur capacité à suivre efficacement les packages de logiciels open source qui contiennent des dépendances transitives liées aux modèles ML présente un angle mort important. Lorsque ces lacunes critiques sont présentes dans le processus de développement logiciel ML, il devient non seulement plus difficile pour les organisations de gérer efficacement leurs chaînes d'approvisionnement en IA/ML, mais cela accroît également le risque de failles de sécurité.



Excès de confiance dans la sécurité de l'IA

Alors que 79 % des organisations déclarent avoir recours à un certain niveau d'analyse des modèles, les solutions actuelles pour la sécurité des modèles IA/ML en sont encore à leurs balbutiements avec des techniques de détection naïves. Par exemple, les approches actuelles ont conduit à un 96 % de faux positifs dans l'identification des modèles malveillants sur Hugging Face, tout en passant à côté de menaces en raison de techniques d'analyse simplistes. Alors que de nombreux outils de sécurité cherchent à tirer parti de la couverture des artefacts de modèles, les organisations doivent évaluer sérieusement l'efficacité des solutions de sécurité proposées par les fournisseurs.

Méthodologie

Ce rapport intègre une combinaison d'informations tirées des données d'utilisation de JFrog, des résultats d'analyse CVE de l'équipe de recherche en sécurité de JFrog et des données d'enquêtes commandées à des tiers. Voici un aperçu plus détaillé de chaque source :

Données d'utilisation de la plateforme JFrog

Ce rapport présente les tendances d'usage technologique issues d'un instantané de fin d'année des données d'utilisation anonymisées de la JFrog Platform for Cloud, reflétant l'activité de milliers de clients, de centaines de milliers de dépôts et de plusieurs pétaoctets de données.

La popularité des packages est représentée par le nombre d'actions (téléchargement), le nombre total d'artefacts, le nombre total de dépôts et la taille totale des artefacts pour un type de package donné. Action Count fournit une bonne représentation de la fréquence à laquelle les différents types de packages sont appelés et générés par les développeurs, en tant qu'indication de l'utilisation réelle dans le développement de logiciels.

Il est possible qu'une poignée d'entreprises faussent ces classements. Cependant, comme nous examinons également les actions des artefacts, nous pouvons conclure avec certitude quel type de package est activement utilisé dans le cadre du processus de développement.

Analyse de l’équipe de recherche en sécurité de JFrog

En tant qu’autorité CNA, l’équipe de recherche en sécurité de JFrog surveille activement les nouvelles vulnérabilités, les analyses pour en déterminer la gravité réelle, et publie ensuite ces renseignements au bénéfice de la communauté et de tous les clients JFrog.

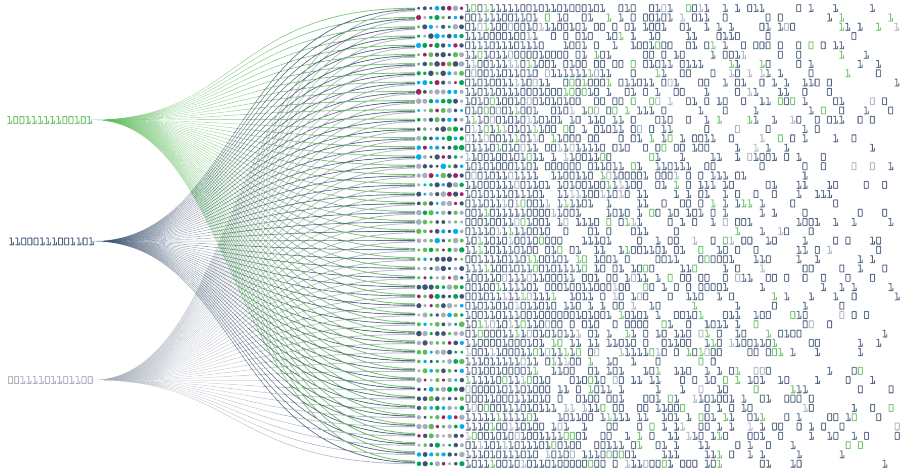
Ce rapport comprend des données tirées de sources publiques via le service JFrog Catalog, des informations CVE tirées de la National Vulnerability Database et des analyses exclusives réalisées par l’équipe de recherche en sécurité de JFrog sur ces sources de données.

Résultats de l’enquête commanditée

JFrog a chargé Atomik Research de mener une enquête internationale en ligne auprès de 1 402 répondants travaillant dans des secteurs d’activité sélectionnés[1] aux États-Unis (n=375), au Royaume-Uni (n=205), en Inde (n=206), en Allemagne (n=205), en France (n=205) et en Israël (n=205). L’échantillon est composé de salariés à temps plein qui occupent des fonctions spécifiques[2] au sein des départements de technologie de l’information, de systèmes d’information ou de technologie de leur organisation.

En outre, toutes les personnes interrogées indiquent que leur entreprise compte au moins 1 000 employés et confirment la présence d’une équipe de développement de logiciels d’au moins 50 membres au sein de leur entreprise. Tous les participants avaient la possibilité d’accéder aux traductions anglaise, française, allemande, hébraïque et hindi du questionnaire en ligne.

La marge d’erreur pour l’ensemble de l’échantillon est de +/- 3 points de pourcentage avec un niveau de confiance de 95 %. L’enquête sur le terrain s’est déroulée entre le 22 novembre et le 9 décembre 2024. Atomik Research est une agence d’études de marché créative.



1. Pour pouvoir participer, tous les répondants devaient indiquer être employés par une organisation qui sert les industries suivantes : (a.) aérospatiale (b.) architecture et ingénierie (c.) automobile (d.) banque, services financiers, assurance et fintech (e.) énergie, pétrole, gaz (f.) gouvernement ou secteur public (g.) santé et sciences de la vie (h.) hôtellerie (i.) production industrielle (j.) commerce de détail (k.) technologie (l.) transport et logistique (m.) services publics, télécom et énergie

2. Pour pouvoir participer, les personnes interrogées doivent déclarer occuper des postes correspondants ou semblables aux fonctions suivantes : (a.) Spécialiste ou ingénieur en IA (b.) ingénieur en sécurité des applications (d.) ingénieur en cybersécurité (e.) data scientist (f.) développeur (g.) Architecte DevOps (h.) Ingénieur DevOps (i.) responsable de l’ingénierie (j.) spécialiste en Machine Learning ou ingénieur ML (k.) ingénieur de plateforme (l.) architecte en sécurité (m.) chercheur en sécurité (n.) ingénieur en fiabilité des sites (o.) architecte logiciel (p.) développeur logiciel (q.) ingénieur logiciel (r.) architecte de solutions en plus d’indiquer un emploi dans les technologies de l’information, les systèmes d’information, les départements technologiques ou le département de développement de produits informatiques de leur organisation.

À propos de la plateforme JFrog

La plateforme JFrog est une solution cloud-native ouverte et hautement évolutive qui s'intègre aux technologies et aux outils de la chaîne d'approvisionnement logicielle. Elle offre aux organisations un contrôle et une traçabilité complets lorsque les composants logiciels passent des développeurs à tous les environnements de déploiement, y compris les modèles ML, les appareils en périphérie et les centres de données de production.

Le présent rapport de données comporte des déclarations dites « prévisionnelles », telles que définies par la réglementation fédérale américaine sur les titres, y compris, entre autres, celles relatives aux données d'utilisation de JFrog et à la chaîne d'approvisionnement logicielle.

Ces déclarations prévisionnelles reposent sur nos hypothèses, attentes et convictions actuelles et sont soumises à des risques, incertitudes, hypothèses et changements de circonstances substantiels susceptibles d'entraîner une différence matérielle entre les résultats, performances ou réalisations réels de JFrog et ceux exprimés ou sous-entendus dans toute déclaration prévisionnelle.

Un nombre important de facteurs pourraient entraîner des écarts significatifs entre les résultats, performances ou réalisations réels et les déclarations faites dans le présent communiqué de presse, y compris, mais sans s'y limiter, les risques présentés dans nos documents déposés auprès de la Securities and Exchange Commission, notamment notre rapport annuel sur le formulaire 10-K pour l'exercice clos le 31 décembre 2024, nos rapports trimestriels sur le formulaire 10-Q, ainsi que tout autre rapport ou document que nous pourrions déposer ponctuellement auprès de la Securities and Exchange Commission. Les déclarations prévisionnelles représentent nos convictions et nos hypothèses uniquement à la date du présent communiqué de presse. Nous déclinons toute obligation de mettre à jour les déclarations prospectives.