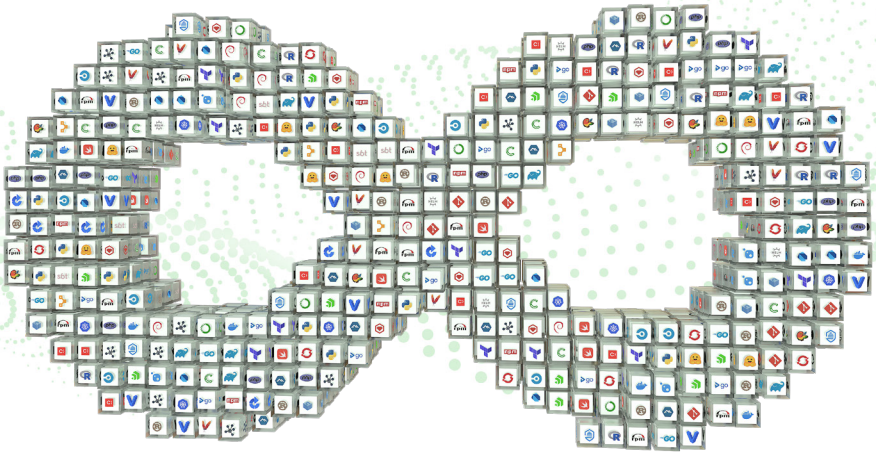
Software-Lieferkette: Zum Stand der Dinge 2025

Wachsende Bedrohungen gefährden die Integrität von Software

Copyright 2025 JFrog Ltd.

Inhaltsverzeichnis

Einführung



Die Verwaltung und Sicherung der gesamten Software-Lieferkette ist eine Grundvoraussetzung für die Bereitstellung vertrauenswürdiger Software-Releases. Das ist jedoch oft leichter gesagt als getan. Als ein auf Softwaresicherheit fokussiertes Unternehmen mit einem eigenen Security-Forschungsteam und mehr als 15 Jahren Erfahrung in der Unterstützung von Development- und Security-Teams kennt JFrog die Bedrohungen und Herausforderungen, mit denen Unternehmen heutzutage konfrontiert sind. Im Zeitalter von KI verschärfen sich diese Herausforderungen zunehmend und die meisten DevSecOps-Teams fragen sich: Wie können wir mit all diesen Veränderungen Schritt halten?

Dieser Report kombiniert JFrog-Nutzungsdaten von Millionen von Usern, CVE-Analysen des JFrog-Security-Forschungsteams und von Drittanbietern erhobene Umfragedaten von 1.400 Sicherheits-, Entwicklungs- und Ops-Experten, um die alles entscheidende Frage zu beantworten. Die daraus resultierende Analyse bietet Einblick in die weite Landschaft der Software-Lieferkette und der Entwicklung, zeigt auf, wo bestehende und neue Risiken liegen und was nötig ist, um Ihre Software-Lieferkette im Jahr 2025 zu sichern.

Wir hoffen, dass dieser Report für Sie von Nutzen ist, und freuen uns über Ihr Feedback an data_report@jfrog.com.

2025 JFrog Ltd. All rights reserved.

Kurzfassung

Die Software-Lieferkette entwickelt sich mit einem noch nie dagewesenen Tempo weiter. Und damit steigt das Risiko, dass Unternehmen in einem kaum noch kontrollierbaren Ausmaß neuen Bedrohungen ausgesetzt sind. Wenn es darum geht, das Risiko in der gesamten Lieferkette zu minimieren, ist „mehr“ nicht zwangsläufig die beste Lösung. Wer schnell handeln, neue Technologie adaptieren und den Wettbewerb dominieren möchte, ist gut beraten, dem Grundsatz “work smarter, not harder” zu folgen – durch die Vereinfachung von Toolchains und Prozessen.

Ihre Software-Lieferkette – größer, schneller, komplexer

Das Open-Source-Ökosystem wächst ungebremst weiter – und Unternehmen, die Innovationen vorantreiben wollen, setzen alles daran, die neuesten Technologien so schnell wie möglich zu integrieren.

- Zwei Drittel der Unternehmen (64 %) geben an, 7 oder mehr Programmiersprachen zu verwenden. 44 % verwenden sogar 10 oder mehr. Im Vergleich zum Vorjahr (53 % bzw. 31 %) ist das ein deutlicher Anstieg.
- Öffentliche Repositorys wachsen weiter. Docker Hub hat im Jahr 2024 unglaubliche 1,9 Millionen Images hinzugefügt, Hugging Face verzeichnet einen Zuwachs von 1 Million Images.
- Ein typisches Unternehmen integriert 458 neue Pakete pro Jahr. Das entspricht durchschnittlich 38 neuen Paketen pro Monat, abhängig von der Anzahl der Developer.

Mehr Risiko, weniger Überblick

Auch wenn das Bewerten der potenziellen Auswirkungen einer Sicherheitslücke (Common Vulnerabilities and Exposures = CVE) nach wie vor ein komplexes Unterfangen ist, stellt es nur die Spitze des Risiko-Eisbergs dar.

- Ein massiver Rückstau bei der NVD (National Vulnerability Database) konnte die Entdeckung neuer Schwachstellen nicht aufhalten. Im Jahr 2024 wurden über 33.000 neue CVEs gemeldet – ein Anstieg um 27 % im Jahresvergleich.
- Das JFrog-Security-Forschungsteam entdeckte 25.229 Exposed Secrets/Tokens in öffentlichen Registrys (ein Anstieg um 64 % gegenüber dem Vorjahr). Davon waren 6.790 aktiv.
- In einer tiefgehenden Analyse von 183 auffälligen CVEs stellte das JFrog-Security-Forschungsteam fest, dass 63 davon in den gescannten Anwendungen von JFrog-Cloud-Kunden niemals ausnutzbar sind.

Beim Umgang mit Sicherheitsrisiken gilt: Die Grundlagen nicht vernachlässigen

Unternehmen implementieren Security-Frameworks auf unterschiedlichen Ebenen und setzen immer mehr Sicherheitstools ein – doch einige grundlegende Best Practices bleiben dabei auf der Strecke.

- 71 % der Befragten geben an, dass Entwickler in ihrem Unternehmen, Pakete direkt aus dem Internet herunterladen dürfen.
- 73 % der Unternehmen setzen 7 oder mehr Sicherheitslösungen ein, 49 % nutzen sogar 10 oder mehr. Im Vorjahr lagen diese Werte noch bei 47 % bzw. 33 %.
- Weniger als die Hälfte der Befragten (43 %) gibt an, dass in ihrem Unternehmen Scans sowohl auf Code- als auch auf Binärebene durchgeführt werden.
- 40 % der Befragten haben keinen vollständigen Einblick in die Herkunft von Software, die in der Produktion ausgeführt wird.

Die Einführung von KI nimmt richtig Fahrt auf

Es gibt heute mehr Möglichkeiten als je zuvor, KI-Services in die Produktion zu integrieren. Das bringt jedoch auch neue Herausforderungen mit sich, denen sich Unternehmen stellen müssen.

- Über eine Million neue Modelle und Datensätze wurden dieses Jahr auf Hugging Face veröffentlicht – gleichzeitig stieg die Zahl der bösartigen Modelle um das 6,5-Fache.
- 64 % der Teams nutzen gehostete Modelle, aber fast die Hälfte der Unternehmen hostet Modelle – proprietär oder als Open Source – zumindest teilweise selbst.
- 37 % der Unternehmen verlassen sich derzeit auf manuelle Prozesse, um eine Liste zugelassener Modelle zu kuratieren und zu pflegen und so den Einsatz von Modell-Artefakten zu kontrollieren.

2025 JFrog Ltd. All rights reserved.

Was steckt in Ihrer Software-Lieferkette?

Eine moderne Software-Lieferkette ist global, komplex aufgebaut und integriert verschiedene Technologien und Quellen – jährlich kommen Millionen neuer Pakete und Librarys in den populärsten Tech-Ökosystemen hinzu. Unternehmen für Softwareentwicklung nutzen heute eine noch nie dagewesene Anzahl an Programmiersprachen und deren jeweiliges Paket-Ökosystem. Während etablierte Technologien nach wie vor weit verbreitet sind, bringen Innovationen in altbekannten ebenso wie aufstrebenden Open-Source-Ökosystemen sowohl Chancen als auch Risiken mit sich, die in diesem Report näher beleuchtet werden.

2025 JFrog Ltd. All rights reserved.

Anzahl der in Development-Teams verwendeten Programmiersprachen

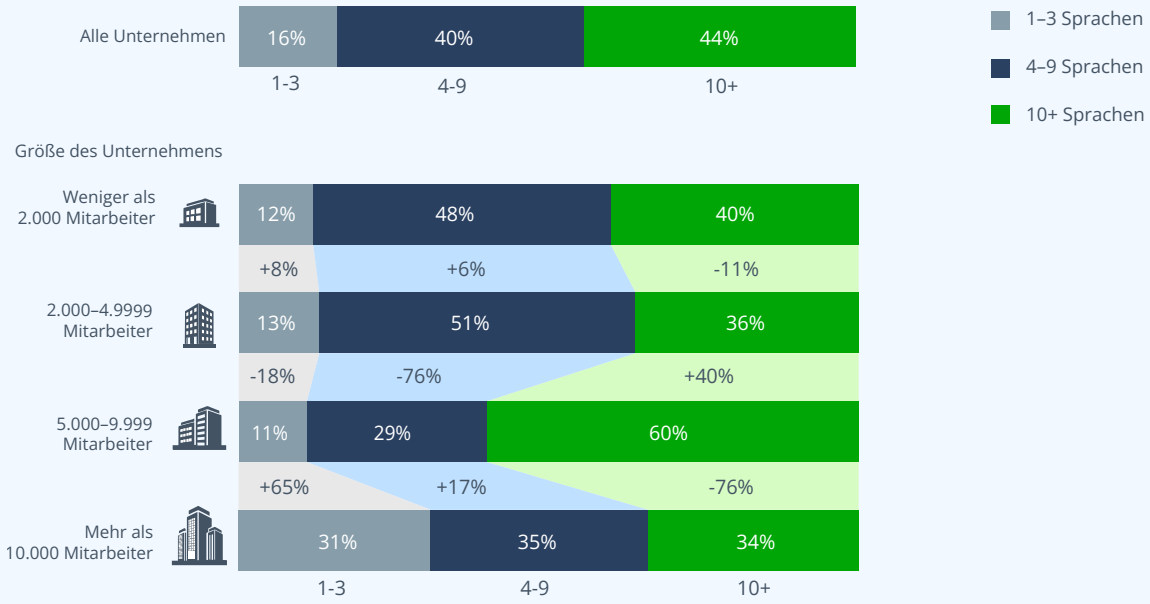


Abbildung 1.1. Wie viele Programmiersprachen verwenden Sie in Ihrem Softwareentwicklungs-Team? (Auftragsstudie, 2024)

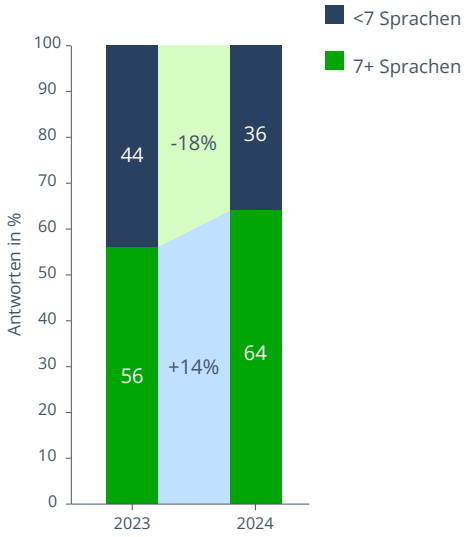


Abbildung 1.2

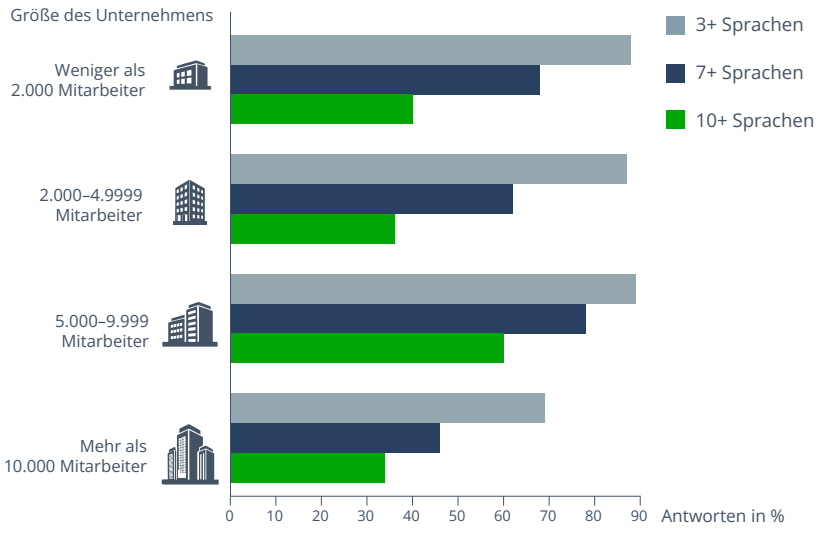


Abbildung 1.3

Fast zwei Drittel der IT-Fachkräfte (64 %) geben an, dass in ihren Unternehmen 7 oder mehr Programmiersprachen zum Einsatz kommen. Im Vorjahr lag dieser Wert bei etwas über der Hälfte der Befragten (56 %). Dieser Anstieg spiegelt die wachsende Komplexität in der gesamten Software-Lieferkette wider.

Mit wachsender Unternehmensgröße steigt in der Regel – wenig überraschend – auch die Anzahl der verwendeten Sprachen. Sobald ein Unternehmen jedoch mehr als 10.000 Mitarbeiter beschäftigt, geht die Anzahl der eingesetzten Sprachen wieder zurück. An diesem Punkt zeichnet sich offenbar ein Kurswechsel ab: Unternehmen erkennen, dass sie ihre Entwicklungsprozesse proaktiver verwalten und durch die Standardisierung bestimmter Technologien den Wildwuchs eindämmen sollten. Möglich ist auch, dass größere Unternehmen etablierte Legacy-Anwendungen beibehalten und weniger neue Projekte realisieren, die den Einsatz zusätzlicher Technologie-Ökosysteme erfordern würden.

2025 JFrog Ltd. All rights reserved.

Neue Pakete nach Jahr und Pakettyp

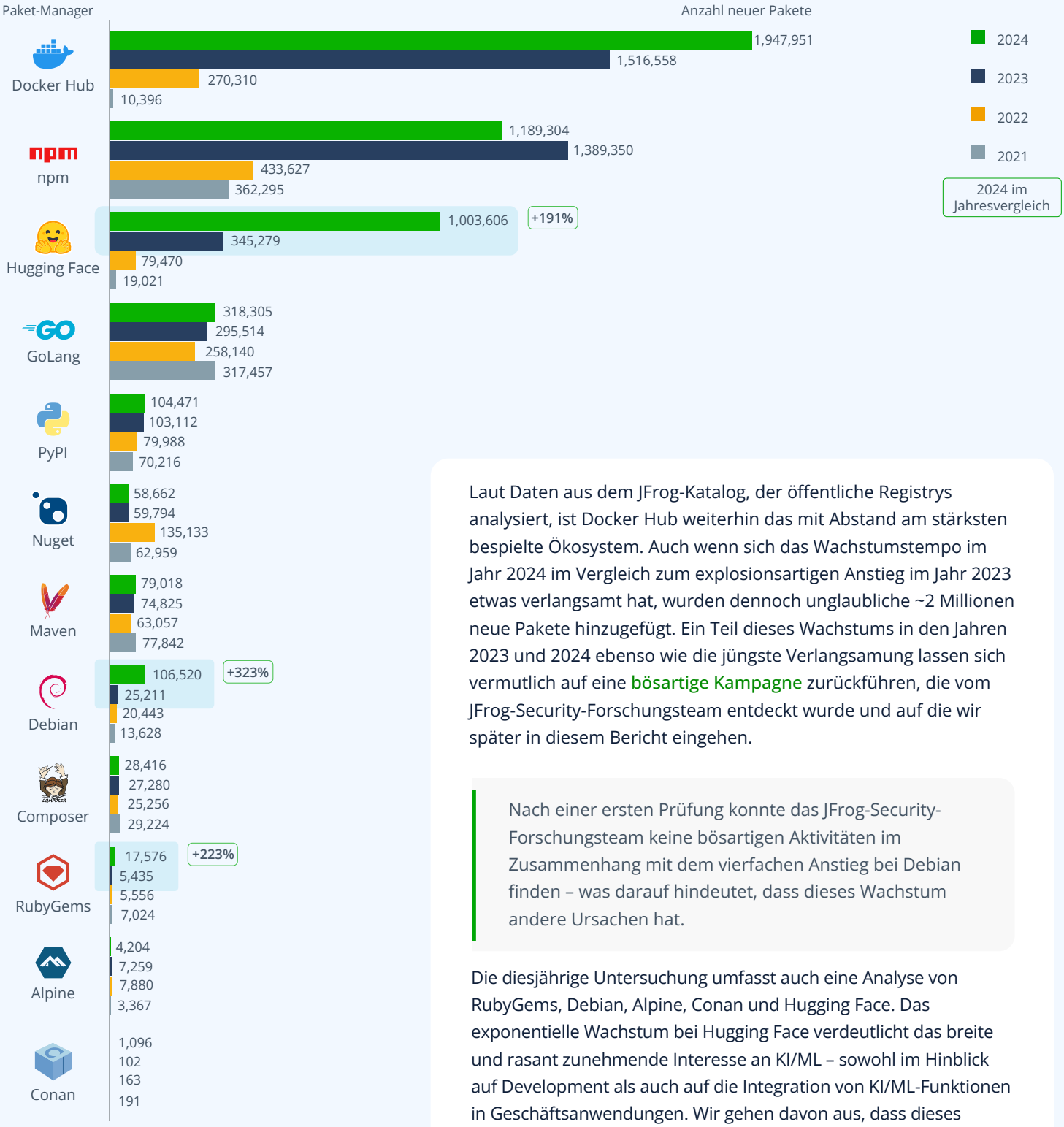


Abbildung 2. Anzahl der neuen Pakete pro Jahr, sortiert nach Pakettyp (JFrog-Katalogdatenbank, 2024)

Laut Daten aus dem JFrog-Katalog, der öffentliche Registrys analysiert, ist Docker Hub weiterhin das mit Abstand am stärksten bespielte Ökosystem. Auch wenn sich das Wachstumstempo im Jahr 2024 im Vergleich zum explosionsartigen Anstieg im Jahr 2023 etwas verlangsamt hat, wurden dennoch unglaubliche ~2 Millionen neue Pakete hinzugefügt. Ein Teil dieses Wachstums in den Jahren 2023 und 2024 ebenso wie die jüngste Verlangsamung lassen sich vermutlich auf eine bösartige Kampagne zurückführen, die vom JFrog-Security-Forschungsteam entdeckt wurde und auf die wir später in diesem Bericht eingehen.

> Nach einer ersten Prüfung konnte das JFrog-Security-Forschungsteam keine bösartigen Aktivitäten im Zusammenhang mit dem vierfachen Anstieg bei Debian finden – was darauf hindeutet, dass dieses Wachstum andere Ursachen hat.

Die diesjährige Untersuchung umfasst auch eine Analyse von RubyGems, Debian, Alpine, Conan und Hugging Face. Das exponentielle Wachstum bei Hugging Face verdeutlicht das breite und rasant zunehmende Interesse an KI/ML – sowohl im Hinblick auf Development als auch auf die Integration von KI/ML-Funktionen in Geschäftsanwendungen. Wir gehen davon aus, dass dieses Ökosystem in diesem Jahr und darüber hinaus weiter wachsen wird.

2025 JFrog Ltd. All rights reserved.

Top-Pakettechnologien, die in Unternehmen genutzt werden

Pakettyp	Requests*	Anzahl der Repositorys	Artefakte
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

*% der Gesamtanfragen aus 57 Milliarden Requests im 4. Quartal

In diesem Jahr haben wir zum Jahresende (Q4) eine Momentaufnahme erstellt, um einen vertieften Einblick in die beliebtesten Technologien unter den 35+ von JFrog unterstützen Technologietypen zu erhalten. Während etablierte Technologie-Ökosysteme wie npm, Docker und Maven weiterhin stark vertreten sind, verzeichneten YUM und Cargo auffällige Popularitätssprünge.

In den vergangenen Jahren konnten wir beobachten, wie die Popularität von Cargo kontinuierlich gestiegen ist – vor allem, weil Behörden zunehmend auf speichersichere Entwicklung drängen. Ob die Beliebtheit von Rust irgendwann stagniert oder die weite Verbreitung und Akzeptanz etablierterer Sprachen wie Java erreichen wird, bleibt abzuwarten.

Erwähnenswert ist auch die zunehmende Nutzung von OCI und Helm-OCI. JFrog führte Anfang 2024 spezielle Repositorys für OCI ein. Viele unserer Kunden nutzen dieses Angebot bereits. Das deutet auf eine wachsende Präferenz für offene Standards für Container und andere Technologie-Ökosysteme hin und ist der Grund, warum wir unsere Terraform-Repositorys um native Unterstützung von OpenTofu erweitert haben.

Der Einsatz gängiger Technologien variiert je nach Branche:

- **Unternehmen aus der Automobil- und IoT-Branche** setzen auf Maven (Backend-Anwendungen), npm (Frontend-Anwendungen), Conan (eingebettete Geräte), Docker und PyPI (für KI/ML). Häufig werden viele dieser Technologien in generischen Paketen gebündelt (tar-/zip-Images).
- **Unternehmen im Bereich KI/ML und Robotik** setzen auf PyPI und ML-Modelle aus öffentlichen Repositorys wie Hugging Face und Tensorflow und speichern diese Modelle in Containern oder generischen Paketen (tar/zips). Darüber hinaus nutzen sie gelegentlich auch native Repositorys wie Hugging Face oder das Machine-Learning-Repository von JFrog* für ihre Modelle.
- **Versicherungs-, Finanz- und Einzelhandelsunternehmen** erwenden eine Kombination von Technologien wie Maven, npm und Docker. Mit dem zunehmenden Einsatz von KI/ML beginnen auch sie, PyPI und ML-Modelle zu nutzen, um ihr Angebot zu verbessern und wettbewerbsfähig zu bleiben.

* Jfrogs Machine-Learning-Repository wurde im Januar 2025 eingeführt und ist daher nicht in den Daten dieses Reports enthalten.

Abbildung 3: Verwendete Technologien sowie Anzahl der Aktionen, Zahl der Repositorys und die Gesamtgröße der gespeicherten Artefakte für jede Technologie (JFrog-Datenbank, 2024)

2025 JFrog Ltd. All rights reserved.

Beliebte Librarys

Position	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

Abbildung 4: Top 20 heruntergeladene Pakete für Docker, Maven, PyPI und npm in der JFrog-Cloud (SaaS (JFrog-Datenbank, 2024)

Viele öffentliche Registrys stellen Download-Metriken für die darin enthaltenen Pakete bereit. Diese Metriken können jedoch aus verschiedenen Gründen irreführend sein, etwa weil sie durch clientseitige Paket-Abrufe beim Ausführen von Builds beeinflusst werden. Stattdessen ermittelt unsere Untersuchung, welche Librarys tatsächlich verwendet werden – basierend auf den Anforderungen innerhalb von JFrog-SaaS-Umgebungen, die von Tausenden Kundenkonten genutzt werden.

Bei Docker ist es wenig überraschend, dass sich unter den Top 20 Images die beliebtesten Betriebssysteme und Programmiersprachen befinden – vermutlich sind sie als Basis-Images im Einsatz. Erfreulich ist, dass alle bis auf eines entweder offizielle Docker-Images sind oder von einem verifizierten Publisher stammen. Das deutet darauf hin, dass auf eine regelmäßige Wartung der Images geachtet wird. Bemerkenswert ist, dass auch das offizielle Docker-Image helloworld zu den Spitzenreitern gehört – ein möglicher Hinweis auf eine beachtliche Sammlung von Demos, Proof of Concepts und auf Entwickler, die sich mit Docker vertraut machen, da Containerisierung zunehmend zum Standard in der modernen Softwarebereitstellung wird.

Was Maven-, PyPI- und npm-Pakete betrifft, gab es in den Top 20 der am häufigsten von Unternehmen genutzten Pakete keine Überraschungen. Unklar bleibt allerdings, ob diese Pakete direkt abgerufen und von Softwareentwicklern explizit ausgewählt werden oder als Abhängigkeiten, möglicherweise sogar als transitive Abhängigkeiten (also Abhängigkeiten von Abhängigkeiten), abgerufen werden.

2025 JFrog Ltd. All rights reserved.

Apache Commons Compress beispielsweise belegt Platz 9 der Popularitätsdaten. Bei näherer Betrachtung dieser Library zeigt sich, dass sie eine direkte Abhängigkeit zu Apache Commons IO, Apache Commons Codec, ASM und Apache Commons Lang aufweist – die wiederum die Plätze 2, 3, 4 und 10 einnehmen. Dies verdeutlicht, wie wichtig es ist, ein aktuelles Inventar über alle in einer Anwendung enthaltenen Software-Artefakte zu führen – häufig in Form einer SBOM. So lässt sich jede einzelne Komponente bewerten und besser einschätzen, welche Ausmaße ein Sicherheitsvorfall annehmen würde, wenn eine bestimmte Komponente anfällig, kompromittiert oder in Ihrer Software-Lieferkette nicht mehr verfügbar ist.

Tempo, mit dem neue OSS-Pakete in Unternehmen Einzug halten

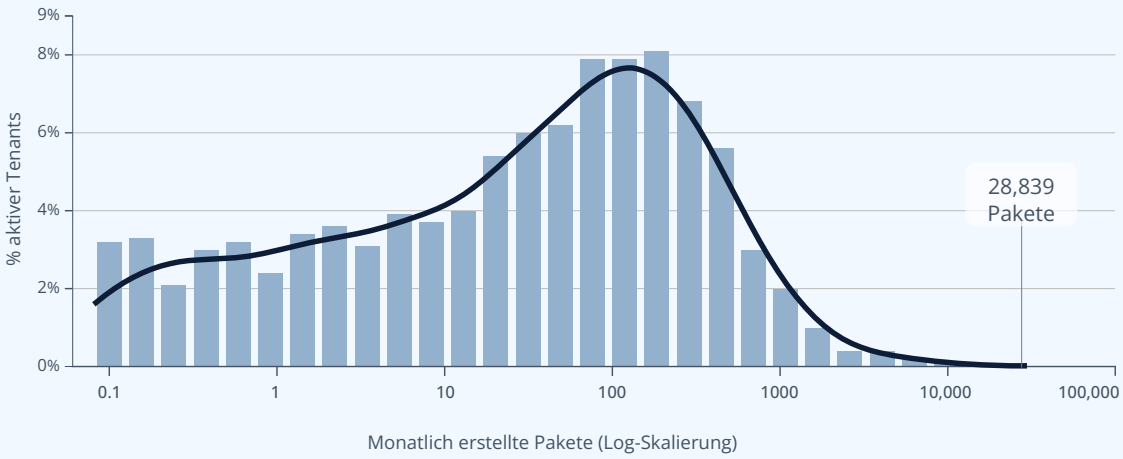


Abbildung 5: Verteilung der monatlich erstellten neuen Pakete durch aktive Tenants im Jahr 2024 (JFrog-Datenbank, 2024)

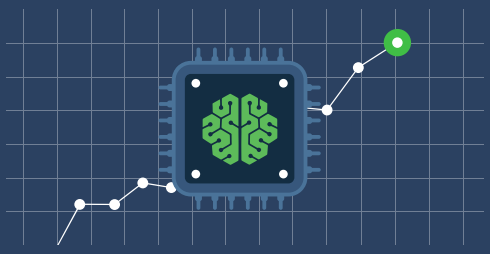
Im Jahr 2024 integrierten Unternehmen, die JFrog Cloud – das Cloud-native SaaS-Angebot von JFrog – nutzen, insgesamt über sieben Millionen neue Pakete in ihre Software-Lieferkette.

Für ein durchschnittliches Unternehmen entspricht das rund 2.000 Paketen pro Jahr – allerdings wird dieser Wert von einigen wenigen besonders aktiven Nutzern nach oben getrieben. Das größte einzelne Unternehmen integrierte im Jahresverlauf ganze 346.000 neue Pakete, während der Median mit vergleichsweise überschaubaren 231 Paketen deutlich niedriger liegt.

Wenn man jene Unternehmen ausklammert, die keine Pakete integriert haben, steigt die durchschnittliche Anzahl der Pakete auf 458 – das entspricht 38 neuen Paketen pro Monat. Auf der Grundlage der Daten liefert diese Zahl vermutlich das realistischste Bild für ein typisches Unternehmen. Selbst ein Tempo von etwas mehr als einem neuen Paket pro Tag kann erhebliche Herausforderungen mit sich bringen – insbesondere in Bezug auf Sicherheit, Betriebsrisiken und Lizenz-Compliance der Komponenten, die in die eigenen Umgebungen integriert werden.

2025 JFrog Ltd. All rights reserved.

Wichtigste Erkenntnisse



KI-Boom

Wir erleben ein exponentielles Wachstum bei der Verfügbarkeit von KI/ML-Komponenten – immer mehr Akteure aus der Open-Source-Community und der Unternehmenswelt bringen sich ein und leisten Beiträge zum Ökosystem (z. B. Nvidia mit der Veröffentlichung von NIM und NVLM). Unternehmen treiben die Integration von KI-Services in ihre Produkte mit hoher Geschwindigkeit voran, was sich an den 500+ Hugging-Face-Repositorys von JFrog-Nutzern zeigt. Umso wichtiger ist es, klare Richtlinien und Strategien dafür zu definieren, wie Open-Source-Modelle und -Datensätze genutzt und gesichert werden – ein Thema, dem wir uns im weiteren Verlauf dieses Reports ausführlich widmen.

Der Schutz von Anwendungen und deren Entwicklung hat höchste Priorität

Die US-Regierung und andere politische Akteure weltweit drängen auf den Einsatz sichererer Programmiersprachen und -frameworks. Auch JFrog verzeichnet in den eigenen Daten einen Anstieg bei der Nutzung von Rust/Cargo. Das deutet darauf hin, dass Unternehmen ihre Anwendungen umgestalten oder neue Projekte von vornherein mit einem stärkeren Fokus auf Sicherheit starten. Außerdem lässt die Beliebtheit von OCI auf die wachsende Besorgnis von Unternehmen schließen, dass bevorzugte Open-Source-Technologien in private Hände übergehen könnten, die kommerzielle Nutzung eingeschränkt wird oder eine Lizenz erforderlich ist.

Risiko verzehnfacht

Zwei Drittel der Unternehmen verwenden 7 oder mehr Sprachen und fast die Hälfte sogar 10 oder mehr. Damit steigt das Risiko für die Unternehmen exponentiell an. Sie müssen sicherstellen, dass sie über eine konsistente Pipeline für eine Vielzahl unterschiedlicher Sprachen, Teams und Bedrohungsquellen verfügen. Jedes Ökosystem hat seine eigenen Schwachstellen, böswillige Akteure und strukturelle Besonderheiten, die bei der Entwicklung berücksichtigt werden müssen – damit Anwendungen in der Produktion auch wirklich sicher sind.

Ein Paket pro Tag? Angreifer in Schach halten

In schnell wachsenden Unternehmen werden täglich ein oder mehrere neue Pakete und Versionen integriert. Das macht automatisierte und optimierte Prozesse zur Gewährleistung der Sicherheit dieser Komponenten in der Software-Lieferkette erforderlich. Da Unternehmen bestrebt sind, ihre Entwicklungsgeschwindigkeit zu erhöhen und Entwickler- sowie Security-Teams dazu zu befähigen, innovative Lösungen für geschäftliche Herausforderungen zu finden, wird auch die Anzahl der neu hinzukommenden Pakete weiter zunehmen.

2025 JFrog Ltd. All rights reserved.

Das wachsende Risiko in Ihrer Software-Lieferkette

Unternehmen befinden sich in einem Wettlauf mit Bedrohungsakteuren und müssen sich mit einigen zentralen Faktoren auseinandersetzen, die keine Anzeichen einer Verlangsamung zeigen:

- CVEs
- Bösartige Pakete
- Open-Source-Lizenzierungsrisiko
- Operationelle Risiken (schlecht verwaltete Pakete, EoL usw.)
- Offenlegung von Secrets
- Fehlkonfigurationen/ menschliches Versagen

Insgesamt zeigt die Analyse, dass die derzeit von Entwicklern und Sicherheitsexperten eingesetzten Tools in manchen Fällen hilfreich sind, in anderen jedoch eher schaden. So können beispielsweise KI-Coding-Assistenten, wenn sie nicht richtig eingesetzt werden, potenziell negative Auswirkungen haben – insbesondere bei schlecht oder unzureichend konfigurierten Funktionen.

Jegliche Jahresvergleiche von CVSS-Werten (Common Vulnerability Scoring System) und CWE-Informationen (Common Weakness Enumeration), die in diesem Abschnitt dargestellt werden, sind in diesem Jahr verzerrt. Grund dafür ist ein mehrmonatiger Zeitraum, in dem die National Vulnerability Database (NVD) nicht in der Lage war, neu entdeckte CVEs (Common Vulnerabilities and Exposures) zu analysieren und ihnen Eigenschaften zuzuweisen. Das führte zu einem erheblichen Rückstau.

Dieser NVD-Rückstau ist ein abschreckendes Beispiel und macht ein anhaltendes Problem in unserer Branche deutlich. Mit der stetig wachsenden Zahl an Librarys – und damit auch an CVEs – müssen Unternehmen Wege finden, dieses zunehmende Risiko nachhaltig zu managen. Hinzu kommt, dass angesichts der aktuellen politischen Lage in den USA weder die Integrität noch die Zukunftsfähigkeit der NVD und des National Institute of Standards and Technology (NIST) garantiert ist.

2025 JFrog Ltd. All rights reserved.

Schwachstellen, die in einer bestimmten Technologie oder einem bestimmten Pakettyp gefunden wurden

Im Jahr 2024 meldeten Sicherheitsexperten weltweit knapp 33.000 neue CVEs – ein Anstieg um 27 % gegenüber 2023 und eine Fortsetzung des jährlichen Wachstumstrends von neu entdeckten CVEs. Angesichts der stetig wachsenden Zahl neuer Open-Source-Pakete ist das zwar nicht überraschend, doch das Tempo des CVE-Wachstums (27 % im Vergleich zum Vorjahr) übertrifft sogar die Wachstumsrate der Pakete (24,5 % im Jahresvergleich). Das ist ein Indikator, der nicht auf die leichte Schulter genommen werden sollte.

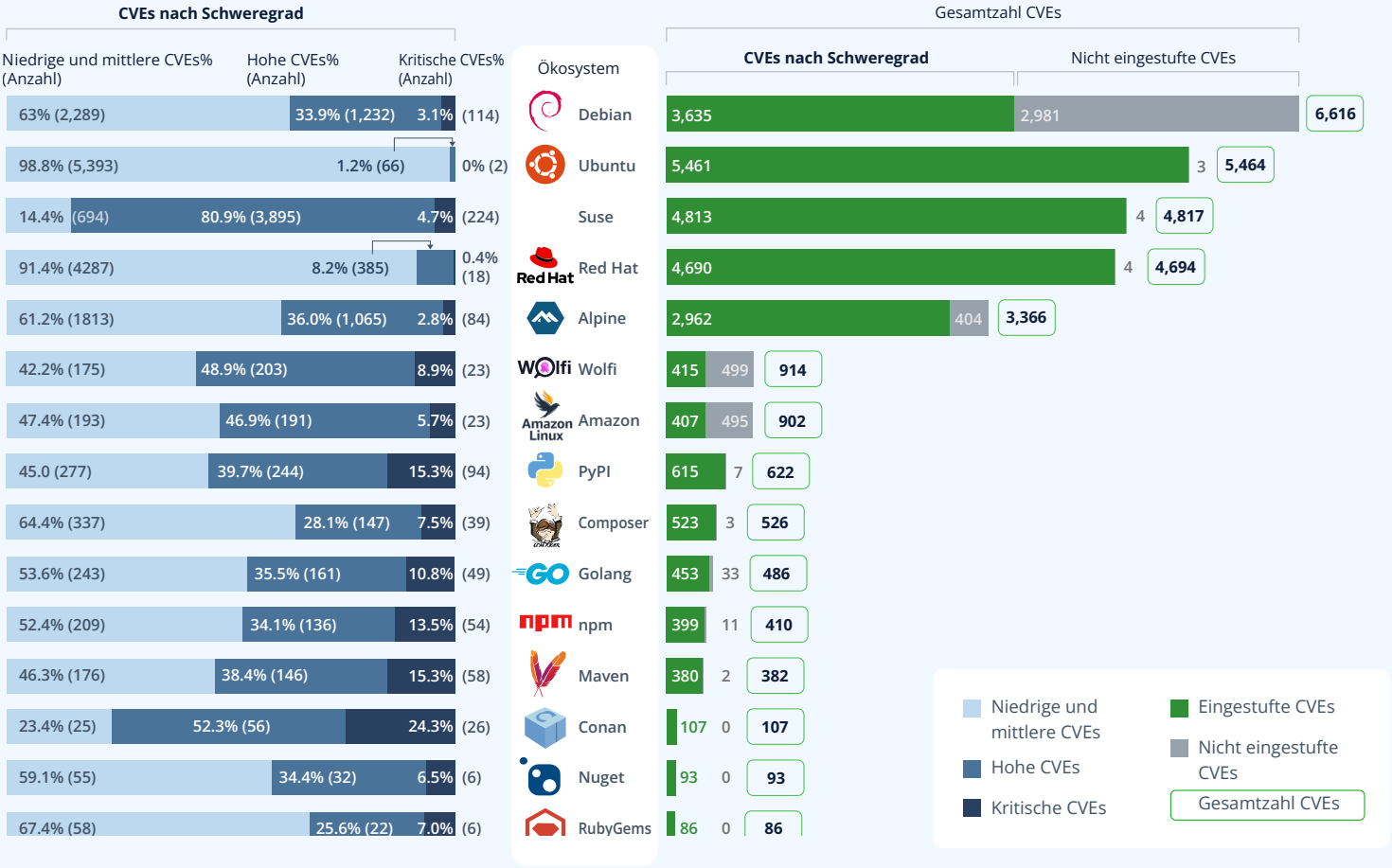


Abbildung 6.1. Anzahl der erkannten CVEs pro Pakettyp im Jahr 2024

Der erste bemerkenswerter Trend ist das starke Wachstum der Debian-CVEs im Jahresvergleich. Angesichts der Vervierfachung der Anzahl der Pakete, die 2024 in das Ökosystem eingebracht wurden, ist dieser Zuwachs nicht überraschend. Erfreulicherweise enthalten die Debian-CVEs des Jahres 2024 nur wenige CVEs mit kritischem oder hohem Risiko.

Ähnlich wie in den Daten für das Jahr 2023 weisen Maven, npm, PyPI und Conan (ein Neuzugang in diesem Jahr) den höchsten Prozentsatz kritischer CVEs auf. Dabei ist die Gesamtzahl der CVEs bei Maven und npm im Vergleich zum Vorjahr zurückgegangen. Ein Blick auf die gesamte Datenbank zum Jahresende zeigt jedoch, dass das bestehende Risiko weiterhin hoch ist – insbesondere bei npm und Maven, in etwas geringerem Maße auch bei PyPI.

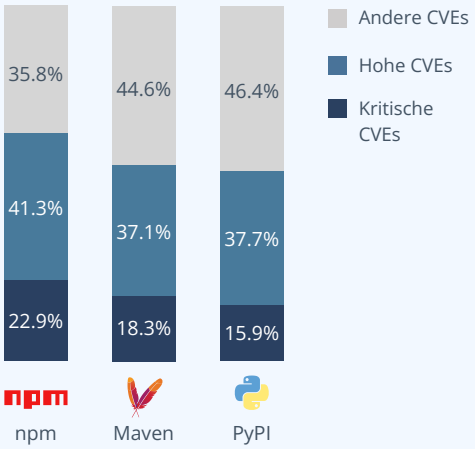


Abbildung 6.2. Kritische und hohe CVEs in populären Ökosystemen zum Jahresende 2024

2025 JFrog Ltd. All rights reserved.

Gesamtzahl entfernter und veralteter Pakete

* Für das Jahr wurden keine Daten erhoben – keine Veränderung

Abbildung 7. Gesamtzahl entfernter und veralteter Pakete (JFrog-Datenbank, 2024)

Pakete werden nicht nur zu Technologie-Ökosystemen hinzugefügt, sondern gelegentlich auch entfernt. Besonders auffällig in diesem Datensatz ist die Zahl der gelöschten Composer-Pakete im Jahr 2024 im Vergleich zu 2023. Eine manuelle Überprüfung durch das JFrog-Security-Forschungsteam ergab die folgenden möglichen Erklärungen:

- Bei den meisten gelöschten Paketen ist das zugehörige GitHub-Repository als „nicht verfügbar“ gekennzeichnet. Das bedeutet, dass sie vom Autor gelöscht oder auf privat gesetzt wurden.
- In einem Fall stellte sich heraus, dass das gelöschte Paket lediglich umbenannt wurde. Möglicherweise kennzeichnet “packagist” einen solchen Umbenennungsvorgang als „gelöscht“.
- Einige Pakete werden gelöscht und als „abandoned“ markiert. Allerdings ist unklar, nach welchen Kriterien ein Paket als “abandoned” gilt.
- Es scheint, als hätte “packagist” im Jahr 2024 eine neue Automatisierung ausgeführt, die Pakete mit ungültigen GitHub-Repositorys gelöscht hat.

Unsere Datenquellen unterscheiden sich in der Art und Weise, wie gelöschte Pakete gemeldet werden; einige stellen diese Informationen bereit, andere nicht. In bestimmten Ökosystemen analysieren wir alle verfügbaren Daten und vergleichen sie über einen längeren Zeitraum hinweg. Doch diese Methode erfasst keine Pakete, die bereits vor unserer ersten Datenerhebung gelöscht wurden. In anderen Fällen erhalten wir regelmäßige Updates, die Änderungen seit unserem letzten Durchlauf enthalten – wobei nur ein paar Ökosysteme in diesen Updates auch Löschungen melden. Entsprechend liegen uns nicht immer vollständige Informationen über gelöschte Pakete vor.

2025 JFrog Ltd. All rights reserved.

Die häufigsten Arten von Schwachstellen

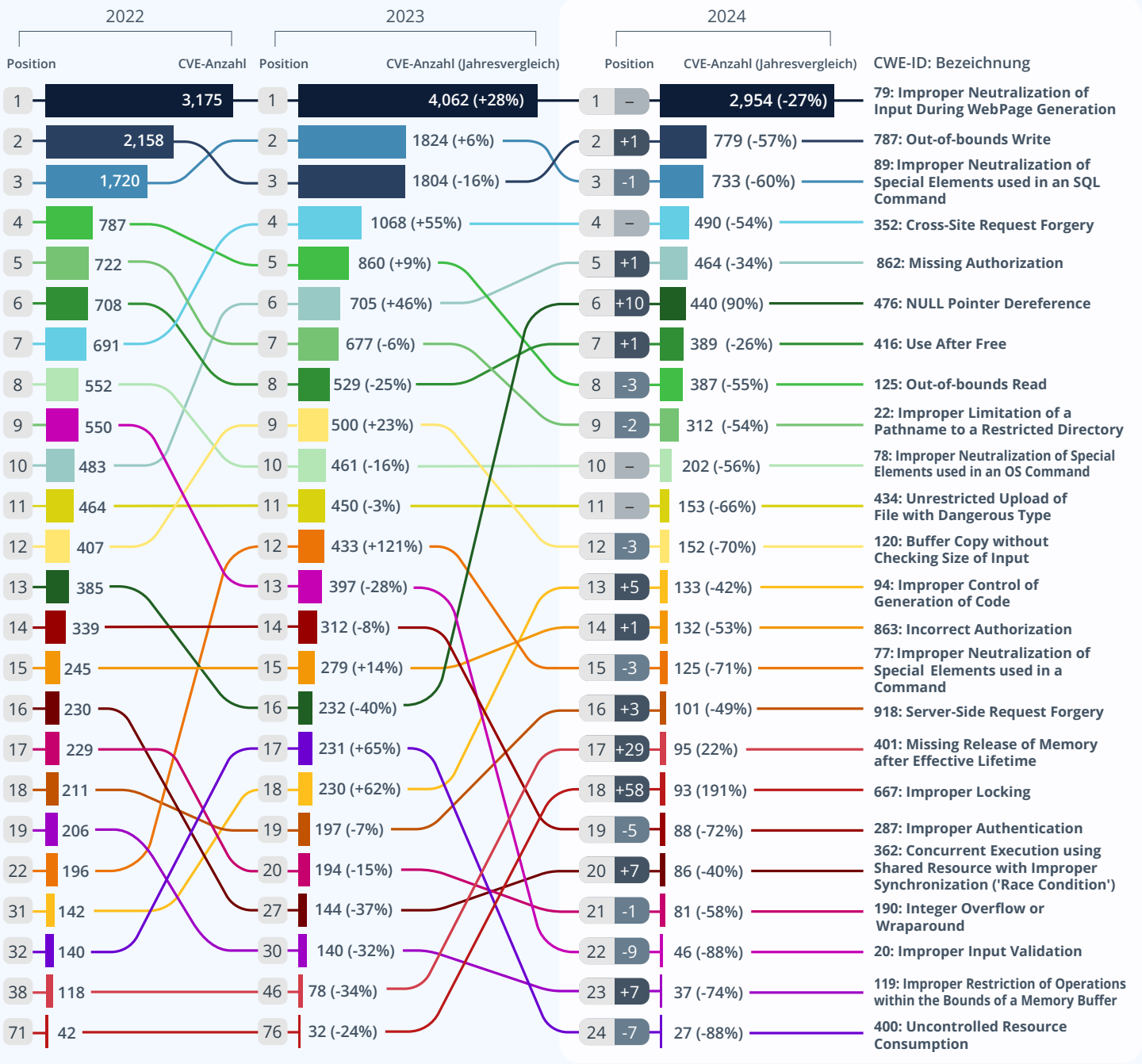


Abbildung 8. Häufige Schwachstellen, die im Jahr 2024 im Vergleich zu 2023, 2022 und 2021 gemeldet wurden

Im Jahr 2024 wurden Schwachstellen (CVEs) 243 unverwechselbare CWE-IDs zugewiesen. Die drei Spitzenreiter sind im Jahresvergleich unverändert Cross-Site-Scripting, Out-of-Bounds-Write und SQL-Injection. In den Top 20 der am häufigsten aufgetretenen Schwachstellen gab es allerdings drei Neuzugänge, die jeweils einen ungewöhnlich starken Anstieg verzeichneten:

401: Missing Release of Memory after Effective Lifetime

2024 #17 ↑ 2023 #46

362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')

2024 #20 ↑ 2023 #28

119: Improper Restriction of Operations within the Bounds of a Memory Buffer.

2024 #23 ↑ 2023 #30

2025 JFrog Ltd. All rights reserved.

Um gegen die drei häufigsten von SAST-Tools erkennbaren CWEs (Cross-Site-Scripting, Out-of-Bounds-Write und SQL-Injection) vorzugehen, empfehlen wir, den Quellcode mit automatisierten SAST-Tools zu scannen, um neue Schwachstellen dieser Art zu verhindern. Darüber hinaus tritt Out-of-Bounds-Write nur bei Low-Level-Programmiersprachen (d. h. speicherunsicheren Sprachen) wie C/C++ auf. Solche Schwachstellen lassen sich durch den Wechsel zu High-Level-Sprachen, wie von der US-Regierung empfohlen, vermeiden.

Es ist wichtig, zu beachten, dass die Jahrestrends bei CWE-Kategorien durch zufällige Faktoren und einmalige Ereignisse verzerrt werden können. So beeinflusst der aktuelle Rückstau bei der NVD mit hoher Wahrscheinlichkeit die Darstellung der CWE-Prävalenz im Jahr 2024. Sobald alle CVEs vollständig erfasst und klassifiziert sind, werden sich diese Zahlen voraussichtlich noch ändern. Eine Betrachtung über einen längeren Zeitraum von beispielsweise 20 Jahren würde aussagekräftigere Trends aufzeigen, da die Verbreitung von Low-Level- gegenüber High-Level-Sprachen Auswirkungen auf die Anzahl der Schwachstellen durch Speicherfehler im Vergleich zu High-Level-/webbasierten Problemen hat. Auch die schwankende Popularität bestimmter Technologien, die jeweils für bestimmte Arten von CWEs anfällig sind, kann diese Trends beeinflussen.

Häufige Auswirkungen von High-Profile-CVEs im Jahr 2024

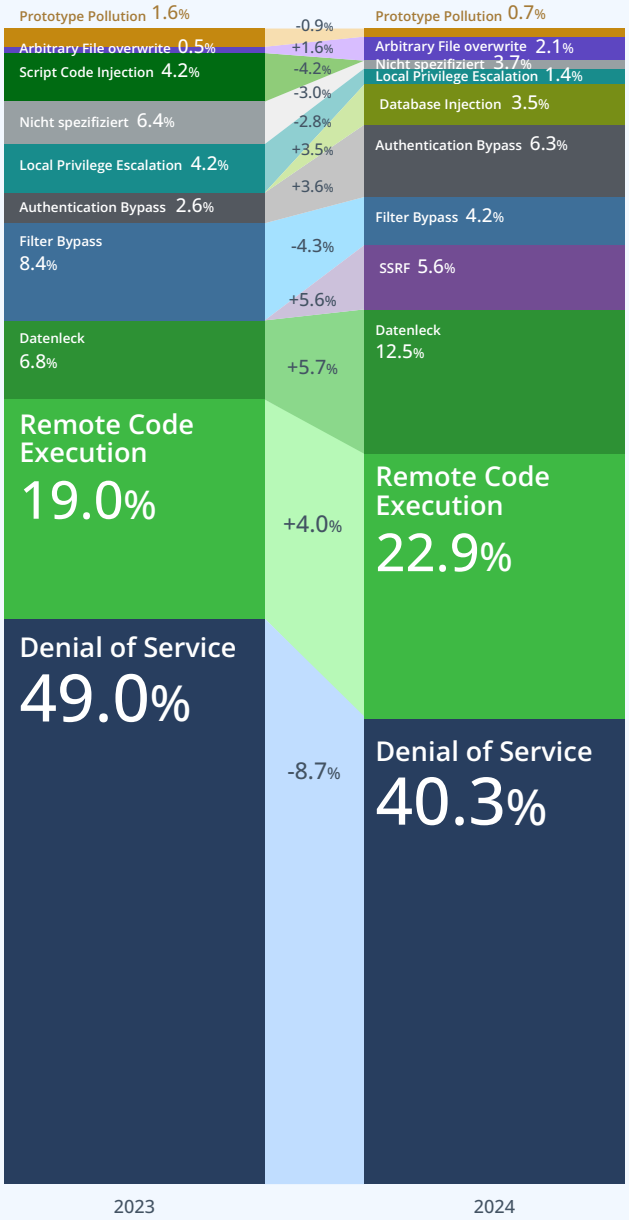


Abbildung 9. Häufige Auswirkungen von High-Profile-CVEs in den Jahren 2023 und 2024

In diesem Jahr analysierte das JFrog-Security-Forschungsteam über 140 High-Profile-CVEs (HPCVE) basierend auf ihrer Relevanz und ihrer potenziellen Auswirkung auf JFrog-Kunden. Denial-of-Service blieb die führende potenzielle Auswirkung einer Schwachstellenexposition (58). Remote-Code-Execution (33) war im Jahresvergleich erneut an zweiter Stelle vertreten, stieg jedoch von 18,9 % auf 22,9 %. Besorgniserregend ist insbesondere der steigende Anteil von Remote-Code-Execution unter den HPCVEs, da solche Schwachstellen Hackern potenziell verheerende Kontrolle über Systeme verschaffen kann.

Datenlecks blieben mit 18 Fällen auf Platz drei, verzeichneten jedoch ebenfalls einen Anstieg im Gesamtprozentsatz. Auch bei Authentication-Bypass und SSRF – einer neu identifizierten Schwachstelle in der diesjährigen Analyse – gab es Zuwächse. Filter-Bypass-Schwachstellen hingegen gingen im Vergleich zum Vorjahr zurück.

Bei der Priorisierung der CVEs für die Forschung berücksichtigt das JFrog-Security-Forschungsteam mehrere Faktoren. Das Team fokussiert sich dabei auf Technologien, die für JFrog-Kunden relevant sind, und priorisiert Schwachstellen mit „hohem“ oder „kritischem“ Schweregrad (d. h. einem CVSS-Score ≥ 7,5). Wenn kein CVSS-Score verfügbar ist, kommt eine Machine-Learning-gestützte Schweregradprognose zum Einsatz. Außerdem priorisiert das Team alle Schwachstellen, die aktiv ausgenutzt werden oder hohe Medienpräsenz aufweisen – selbst wenn sie ein mittleres oder niedriges öffentliches Severity-Rating erhalten haben.

2025 JFrog Ltd. All rights reserved.

Schweregrad der Schwachstellen, die in Ihre Software-Lieferkette eingeschleust werden

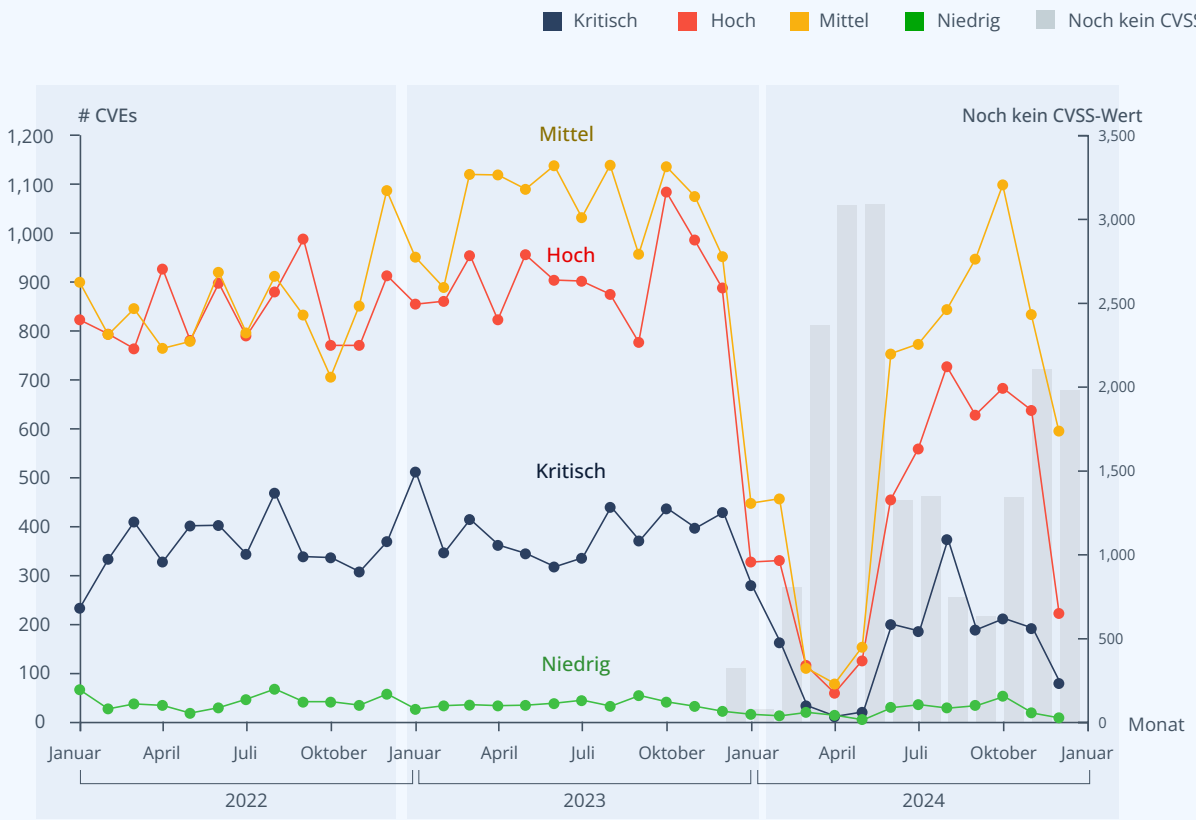


Abbildung 10.1. CVEs nach Monat und Schweregrad in den letzten drei Jahren (National Vulnerability Database)

Die Daten zeigen einen deutlichen Rückgang und anschließenden Aufschwung der CVSS-Zuweisungen Mitte 2024, was jedoch irreführend ist. Der vermeintliche Rückgang ist darauf zurückzuführen, dass die NVD im Februar 2024 im Zuge einer Umstrukturierung aufgrund von Kürzungen die CVE-Forschung eingestellt hat. Um dieses Problem zu lösen, kündigte die NVD im Juni 2024 an, dass die CISA beauftragt wurde, sie bei ihrer Recherche zu unterstützen. Ohne diese Um dem Mangel an Ressourcen entgegenzuwirken, lagert die NVD die CVSS-Bewertung nun größtenteils an externe Partner aus – sogenannte Authorized Data Publishers (ADPs). Derzeit ist CISA der erste und einzige Data Publisher. Das JFrog-Security-Forschungsteam analysiert fortlaufend die Bewertungsmuster der von CISA bewerteten CVEs. Erste Auswertungen deuten darauf hin, **dass CISA sogar noch drastischere Einstufungen (mit höherer Gewichtung des Schweregrads) vergibt als die NVD.** Mit Blick in die Zukunft wirft das auch Fragen hinsichtlich potenzieller Inkonsistenzen bei der CVE-Bewertung auf, wenn weitere autorisierte Datenanbieter online gehen. Unternehmen werden sich dieser neuen Realität stellen müssen, wenn es um die Priorisierung von Sicherheitsmaßnahmen geht.

Basierend auf den verfügbaren NVD-Daten setzt sich der Trend fort: Es gibt weiterhin eine hohe Anzahl von CVEs mit mittlerem und hohem Schweregrad, nur wenige mit niedrigem Schweregrad und die Zahl von CVEs mit kritischem Schweregrad liegt zwischen den Gesamtwerten für niedrig und hoch/mittel.

2025 JFrog Ltd. All rights reserved.

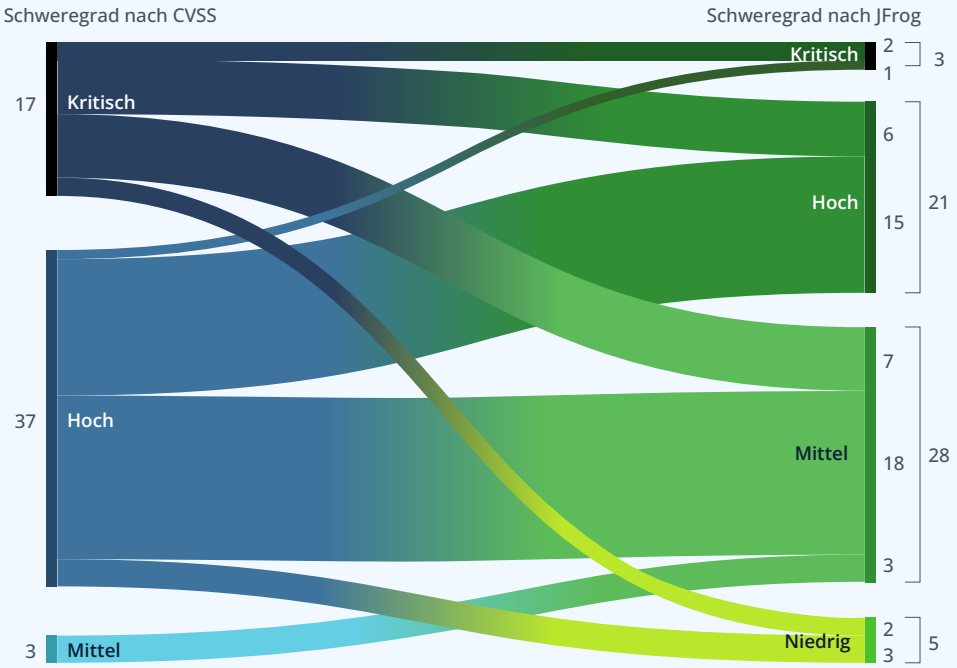


Abbildung 10.2. CVE-Schweregrad (eigene Bewertung durch das JFrog-Security-Forschungsteam im Vergleich zur Schweregrad-Einstufung der NVD)

Allerdings sind nicht alle CVE-Bewertungen das, was sie zu sein scheinen. Das JFrog-Security-Research-Team bewertet regelmäßig CVEs, um ihre tatsächlichen Auswirkungen zu bestimmen, und vergibt ein JFrog Severity Rating. Dieses Rating, das von den DevSecOps-Experten bei JFrog erstellt wird, berücksichtigt auch die Voraussetzungen, die das Ausnutzen der Schwachstellen überhaupt erst ermöglichen. Die CVSS-Ratings dagegen betrachten ausschließlich den Schweregrad des erfolgreichen Exploits einer Schwachstelle und nicht, wie ausnutzbar die Schwachstelle ist.Manchmal ist die Konfiguration oder die Exploit-Methode eine nicht standardmäßige Einstellung für das jeweilige Paket oder die jeweilige Abhängigkeit, wodurch es sehr unwahrscheinlich ist, dass die Schwachstelle jemals ausgenutzt werden kann.

Diese Übergewichtung der CVE-Bewertung ist seit Jahren ein Problem. Besorgniserregend ist diese Tendenz, CVEs höher zu bewerten, deshalb, weil sich die Bewertungsmethode nicht geändert hat. Da Bewertungsmechanismen maßgeblich die erste Risikoeinschätzung eines Pakets beeinflussen, erhöht eine Übergewichtung von CVEs das Potenzial für Fehlalarme deutlich.

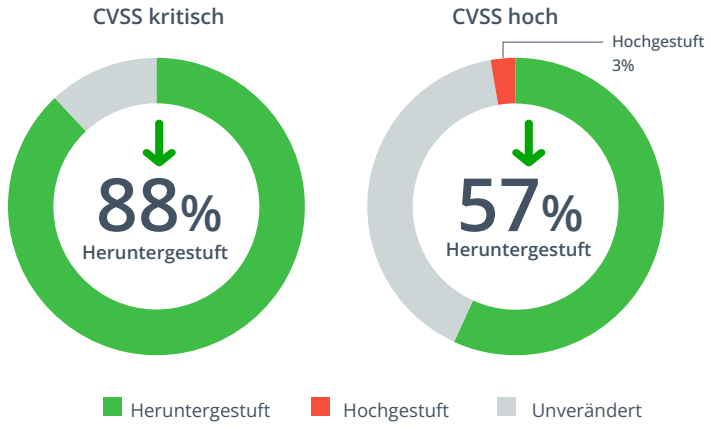


Abbildung 10.3

Basierend auf einer Stichprobe von 140 High-Profile-CVEs zeigte die Analyse des JFrog-Security-Forschungsteams, dass 88 % der als kritisch und 57 % der als hoch eingestuften CVE-Scores in Wirklichkeit weniger schwerwiegend waren, als es die CVSS-Bewertung vermuten lässt.

2025 JFrog Ltd. All rights reserved.

Anwendbarkeits-Rating von High-Profile-CVEs

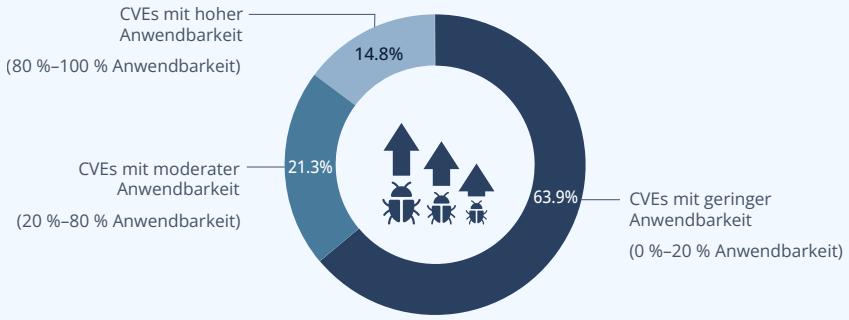


Abbildung 10.4: Anwendbarkeits-Rating von 183 High-Profile-CVEs (eigene Analyse durch das JFrog-Security-Forschungsteam basierend auf CVE- und JFrog-Datenbanken)

Die bloße Zuweisung von Schweregraden zu CVEs reicht nicht aus, um die tatsächlichen Auswirkungen einer Schwachstelle auf ein bestimmtes Softwareprodukt zu beurteilen. Daher geht JFrog Security Research über die Zuweisung von JFrog-Schweregraden hinaus und analysiert zusätzlich die Bedingungen, die die Ausnutzbarkeit dieser Schwachstellen beeinflussen. Zu diesem Zweck entwickelt JFrog „Anwendbarkeits"-Scanner, die prüfen, ob die Voraussetzungen für eine Ausnutzung in einem konkreten Softwareprodukt erfüllt sind.

Das JFrog-Security-Forschungsteam hat Anwendbarkeits-Scanner für 183 im Jahr 2024 veröffentlichte CVEs (CVE-2024-*) entwickelt – mit Fokus auf hohen und kritischen CVEs der von unseren Kunden am häufigsten genutzten Komponenten und Technologien. Die dargestellte Grafik zeigt, wie häufig diese CVEs von JFrog-Kunden als anwendbar (d. h. potenziell von einem böswilligen Akteur ausnutzbar) bzw. als nicht anwendbar (d. h. nicht ausnutzbar) eingestuft wurden. Nur 27 CVEs (15 %) wurden als hochgradig ausnutzbar eingestuft, mit einer Anwendbarkeitsrate von über 80 % in Artefakten, die 2024 von JFrog Xray gescannt wurden. Im Gegensatz dazu zeigen 117 CVEs (64 %) eine geringe Ausnutzbarkeit mit einer Anwendbarkeitsrate zwischen 0–20 %.

CVE-2024-24792 ist ein bemerkenswertes Beispiel mit einer sehr hohen Anwendbarkeitsrate von 99,6 %. Diese Schwachstelle kann durch die typische Nutzung des TIFF-Parsing-Pakets in der Programmiersprache Go ausgelöst werden und tritt häufig in Anwendungen auf, die Image-Uploads und -Verarbeitung unterstützen. Die Schwachstelle ist deshalb so breit anwendbar, da sie getriggert werden kann, wenn die Library verwendet wird, um TIFF-Images zu verarbeiten, die von Usern manipuliert werden können – was in der Folge zu einem Panic im Anwendungscode führen kann.

Im Gegensatz dazu gehört CVE-2024-45490 (eine Schwachstelle in Expat, einem in C geschriebenen XML-Parser) zu den am wenigsten anwendbaren CVEs – in weniger als 10 % der Fälle wurde sie als relevant eingestuft. Um diese Schwachstelle auszunutzen, müsste ein Angreifer den „len"-Parameter manipulieren, der an die API-Funktion XML_ParseBuffer() der Library übergeben wird. Dieses Szenario ist jedoch höchst unwahrscheinlich, da Developer die Länge des XML-Dokuments in der Regel selbst festlegen, häufig über Funktionen wie 'stat' oder 'XML_GetBuffer'.

2025 JFrog Ltd. All rights reserved.

Anwendbarkeitstypen von High-Profile-CVEs

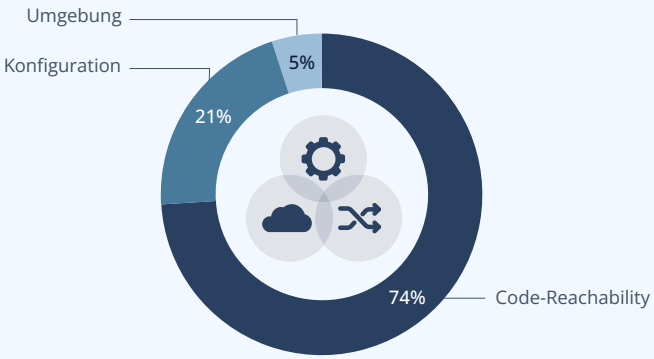


Abbildung 10.5: Anwendbarkeitstypen von CVEs im Jahr 2024
(eigene Analyse durch das JFrog-Security-Forschungsteam basierend auf CVE- und JFrog-Datenbanken)

Das JFrog-Security-Forschungsteam untersuchte außerdem, unter welchen Bedingungen die analysierten CVEs in einer Anwendung erreichbar und ausnutzbar wären. Um festzustellen, ob eine Schwachstelle anwendbar oder ausnutzbar ist, reicht eine Analyse der Reachability (Erreichbarkeit) des anfälligen Codes über eine klassische Call-Reachability-Analyse nicht aus. Es ist ebenso entscheidend, die Konfigurationseinstellungen von Anwendungen und Librarys sowie die Umgebungsbedingungen des zugrundeliegenden Betriebssystems zu untersuchen. Dieser ganzheitliche Ansatz ermöglicht eine umfassende Bewertung potenzieller Risiken, während eine reine Identifizierung von erreichbarem Code wichtige Faktoren außer Acht lässt, die die Ausnutzung von Schwachstellen maßgeblich beeinflussen können.

Ein Beispiel dafür ist die bekannte Schwachstelle „Sudoedit bypass“ (CVE-2023-22809). Die Anwendbarkeit der Schwachstelle lässt sich nur durch die Analyse der Sudo-Konfigurationsdatei „sudoers“ und die Suche nach einer bestimmten, nicht standardmäßigen Konfiguration feststellen. Eine Analyse der Code-Reachability erlaubt in diesem Fall keine Aussage darüber, ob die Schwachstelle anwendbar ist, da die anfällige Komponente „sudo“ ein eigenständiges Dienstprogramm ist und keine Code-Library, die vom Anwendungscode aufgerufen werden kann.

2025 JFrog Ltd. All rights reserved.

Einige Schadpakete sind schlimmer als andere

In unserem Report für das Jahr 2024 haben wir auf die Häufung bösartiger Pakete im npm-Ökosystem hingewiesen. Eine Auswertung der beliebten Paket-Ökosysteme zum Jahresende bestätigt, dass npm das am stärksten betroffene Ökosystem ist, wenn es um die Präsenz bösartiger Pakete geht.Erwähnenswert (und angesichts des rasanten Popularitätsanstiegs nicht überraschend) ist der etwa 6,5-fachen Anstieg an bösartigen Modellen, die in diesem Jahr in das Hugging-Face-Ökosystem hochgeladen wurden. Im Folgenden sind drei auffällige Angriffe aufgeführt, die die Aufmerksamkeit des JFrog-Security-Research-Teams verdient haben:

XZ Utils Backdoor

Am 29. März wurde eine schwerwiegende Sicherheitslücke in XZ Utils gemeldet – einem weit verbreiteten Paket, das in den wichtigsten Linux-Distributionen zum Einsatz kommt. Es enthielt bösartigen Code, der unautorisierten Remote-Zugriff per SSH ermöglichte. Die ausgeklügelte Backdoor, die in den Versionen 5.6.0 und 5.6.1 entdeckt wurde, modifizierte OpenSSH-Serverroutinen, sodass bestimmte Angreifer vor der Authentifizierung beliebige Payloads ausführen und so vollständige Kontrolle über die betroffenen Rechner erlangen konnten.

Quelle >

Docker Hub

Jüngste Malware-Kampagnen auf Docker Hub haben zur Erstellung von Millionen „imageless" Repositorys geführt, die statt Container-Images schädliche Metadaten enthalten. Alarmierend ist, dass fast 20 % (etwa drei Millionen) dieser öffentlichen Repositorys schädliche Inhalte enthielten – von Spam, der Raubkopien bewirbt, über Malware bis hin zu Phishing-Seiten, die von automatisierten Accounts hochgeladen wurden.

Quelle >

Hugging Face

Beim Monitoring von KI-Modellen wurde eine Modellfamilie entdeckt, die beim Laden einer Pickle-Datei automatisch Code ausführt. So erhalten Angreifer über eine Backdoor eine Connectback-Shell und volle Kontrolle über den kompromittierten Computer. Diese stille Infiltration birgt erhebliche Risiken: Sie kann den Zugriff auf kritische Systemen ermöglichen und dadurch großflächige Datenlecks verursachen oder Wirtschaftsspionage begünstigen, ohne dass Betroffene etwas davon mitbekommen.

Quelle >

2025 JFrog Ltd. All rights reserved.

Andere Risikoquellen, die sich in Ihrem Code verbergen

CISOs und AppSec-Teams wissen längst, wie wichtig es ist, genau zu prüfen, was aus der Open-Source-Community übernommen wird. Doch das ist nicht der einzige Bereich, der für eine umfassende Anwendungssicherheit beachtet werden muss.

Fehlkonfigurationen und andere Fehler – die Auswirkungen menschlichen Versagens

Im Jahr 2024 kam es zu einigen Sicherheitsvorfällen, bei denen Daten durch Leaks, Exposures und Fehlkonfigurationen nach außen gedrungen sind.

April 2024

Home Depot war von einer Datenschutzverletzung betroffen, nachdem ein externer SaaS-Anbieter einen Teil der Mitarbeiterdaten veröffentlichte und personenbezogene Daten von 10.000 Mitarbeitern offenlegte.

Quelle >

September 2024

Daten von rund 2.000 Fortinet-Kunden, die auf einer Azure-SharePoint-Website gespeichert waren, wurden von einem Hacker abgerufen und anschließend im Internet veröffentlicht.

Quelle >

August 2024

Tausende Oracle-NetSuite-Kunden leakten durch öffentlich zugängliche Webshops, die mit NetSuite SuiteCommerce oder NetSuite Site Builder erstellt wurden, unbeabsichtigt sensible Daten.

Quelle >

September 2024

In der Low-Code-SaaS-Plattform Microsoft Power Pages wurde aufgrund falsch konfigurierter Zugriffskontrollen ein erhebliches Datenleck entdeckt, von dem möglicherweise Millionen von Usern betroffen sind.

Quelle >

September 2024

Über 1.000 falsch konfigurierte ServiceNow-Enterprise-Instanzen machten Knowledge-Base-Artikel mit sensiblen Unternehmensinformationen für externe Nutzer und potenzielle Bedrohungsakteure zugänglich.

Quelle >

Dezember 2024

Das Datenleck bei Cariad, dem Automotive-Software-Unternehmen von Volkswagen, zählt zu den gravierendsten SaaS-Fehlkonfigurationen des Jahres 2024. Bei dem Vorfall wurden Daten von etwa 800.000 Elektroautos, darunter genaue Fahrzeugstandorte und Informationen, die mit Fahrernamen verknüpft werden könnten, veröffentlicht.

Quelle >

2025 JFrog Ltd. All rights reserved.

Status von geleakten Secrets in Binärartefakten

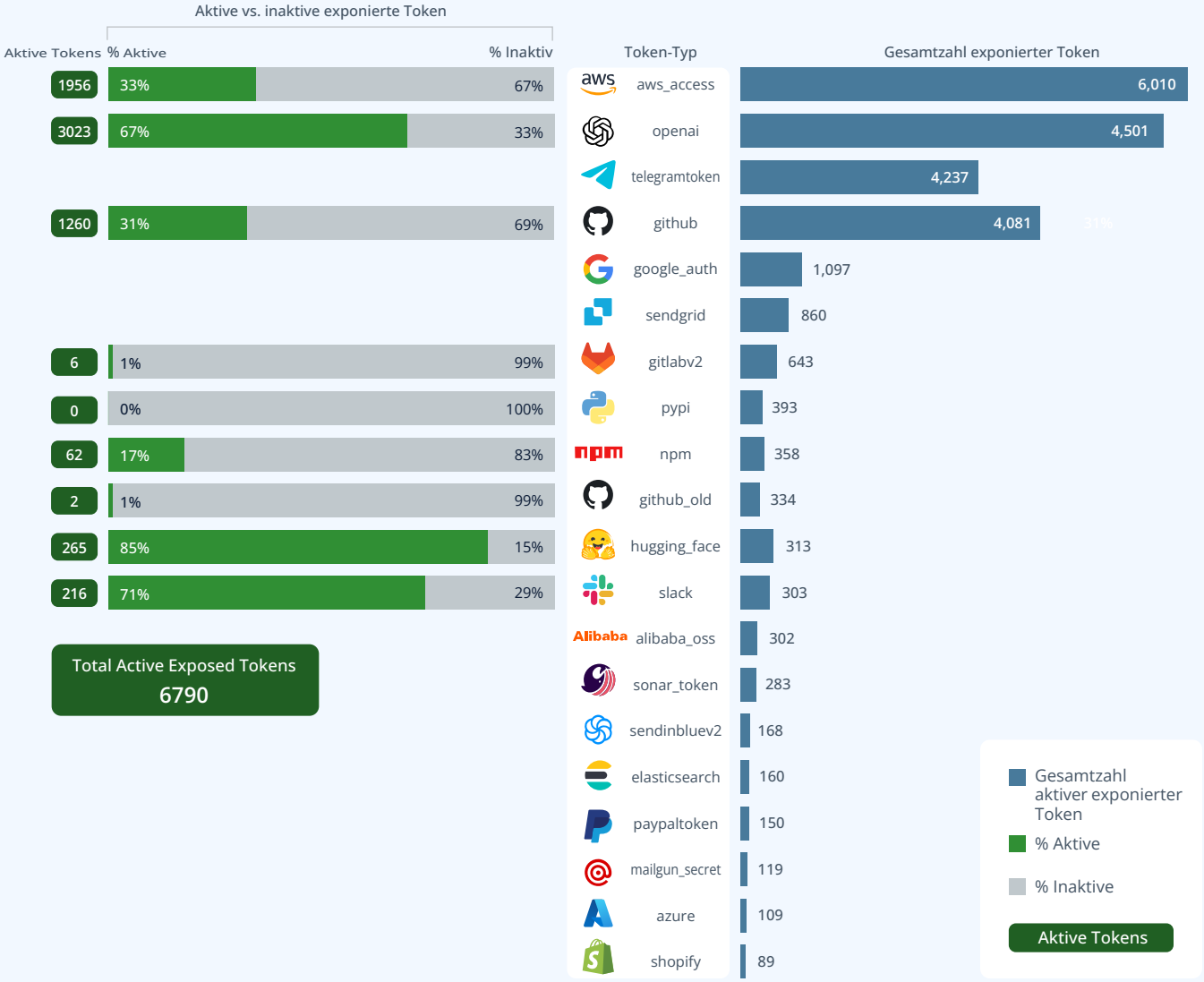


Abbildung 11.1: Die 20 am stärksten exponierten Token-Typen im Jahr 2024

Das JFrog-Security-Forschungsteam hat Millionen von Artefakten in den gängigsten Open-Source-Software-Registrys untersucht: DockerHub, npm und PyPI. In diesem Jahr wurde zudem erfasst, wo aktive Token gefunden wurden (also Token, die zum Zeitpunkt der Datenerhebung tatsächlich nutzbar waren).

Von Jahr zu Jahr gab es Zuwächse bei nahezu allen entdeckten Token-Typen. Insgesamt stieg die Zahl der Exposed Secrets im Vergleich zum Vorjahr um 66 %. Die am stärksten exponierten Token waren dieselben wie in unserem letzten Bericht. Auch sie verzeichneten deutliche Anstiege: AWS stieg um 70 %, OpenAI um 103 %, Telegram um 62 % und GitHub um 82 %. Auch GCP-Token erlebten einen starken Anstieg – mit einem Zuwachs von 86 % im Jahresvergleich.

Hugging-Face-Token sind ein neuer Token-Typ, der in diesem Jahr in die Scanner des JFrog-Security-Forschungsteams aufgenommen wurde – ein klares Zeichen für die wachsende Popularität von Open-Source-Modellen und -Datensätzen. Hugging-Face-Token machen mit rund 85 % den höchsten Anteil aktiver Token unter allen auf der Liste aus. Bemerkenswert ist, dass das JFrog-Security-Forschungsteam 6.790 aktive Secrets zum Zeitpunkt der Datenerhebung identifiziert hat, was böswilligen Akteuren potenziell weitreichenden Zugriff auf proprietäre Systemen ermöglichen könnte.

2025 JFrog Ltd. All rights reserved.

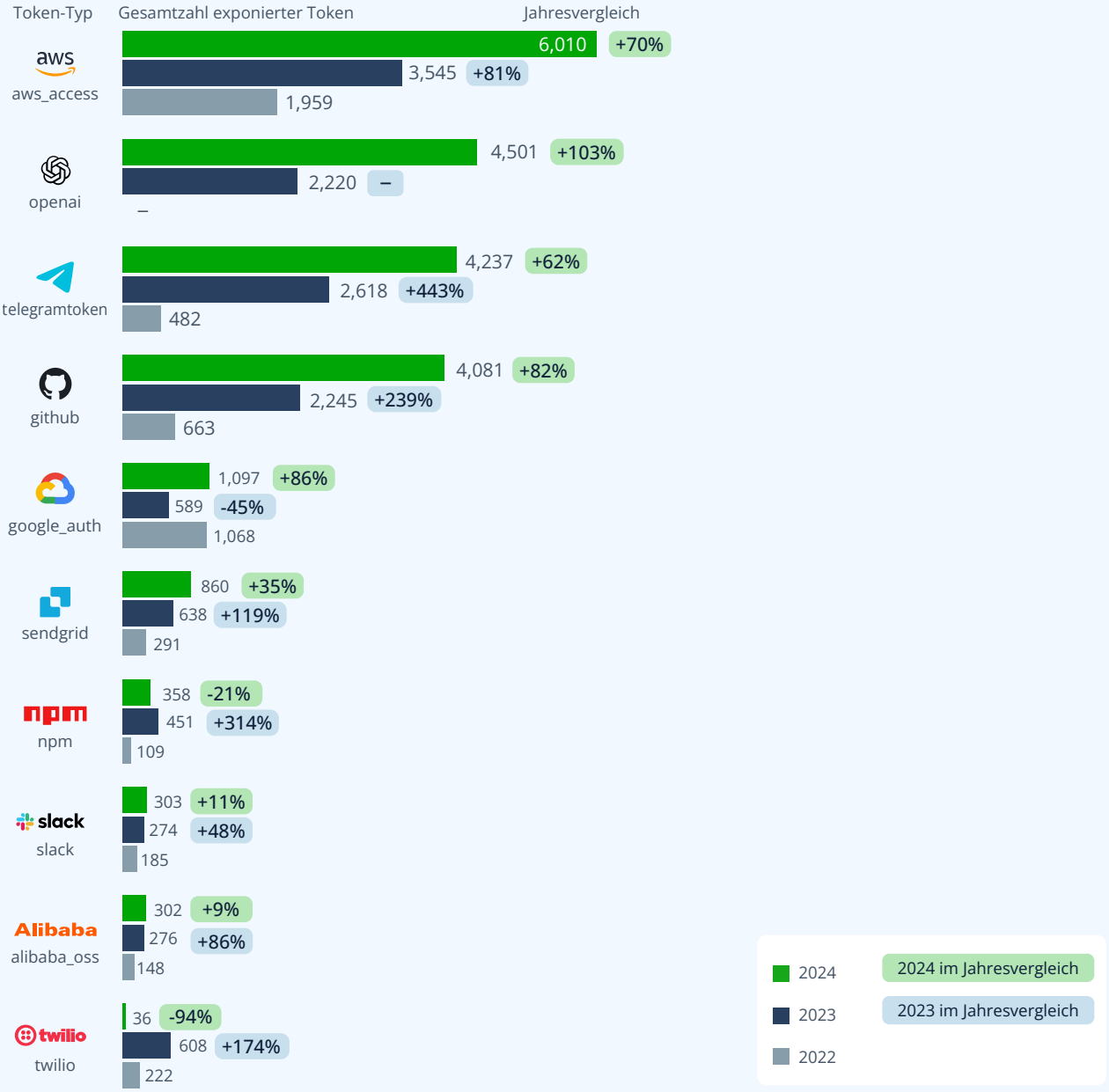
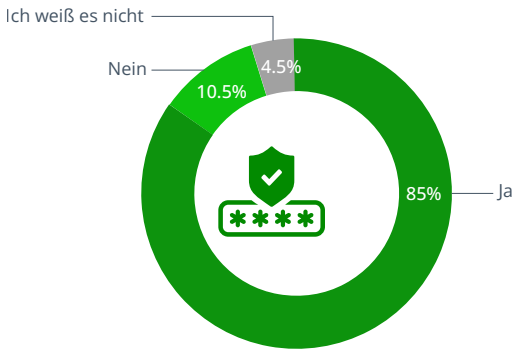


Abbildung 11.2. Jahresvergleich der häufigsten exponierten Token

Verfügt Ihr Unternehmen über Sicherheitsmaßnahmen zur Erkennung von in der Codebasis verbliebenen Secrets und/oder geleakten Token? (Auftragsstudie, 2024)

Unternehmen investieren gezielt, damit ihre Secrets nicht in die Hände böswilliger Akteure oder der Öffentlichkeit gelangen. Allerdings verfügt ein erheblicher Anteil von 15 % der befragten Unternehmen entweder über keine entsprechenden Vorkehrungen oder hat darüber keine Kenntnis. Angesichts der aktuellen Lage rund um geleakte Secrets und der Tatsache, dass bei nahezu allen entdeckten Token-Typen ein Anstieg zu verzeichnen war, ist dieser Anteil besorgniserregend hoch – denn diese Unternehmen setzen sich einem erheblichen Risiko aus.



2025 JFrog Ltd. All rights reserved.

Wie schwerwiegend kann ein Leak von Secrets sein?

Im Juni 2024 entdeckte und meldete das JFrog-Security-Forschungsteam ein geleaktes Zugriffstoken mit Administratorrechten für die GitHub-Repositorys von Python, PyPI und der Python Software Foundation. Das Token war in einem öffentlichen auf Docker Hub gehosteten Docker-Container geleakt worden.

Als Community-Service scannt das JFrog-Security-Forschungsteam kontinuierlich öffentliche Repositorys wie Docker Hub, npm und PyPI, um bösartige Pakete und geleakte Secrets zu identifizieren. Das Team meldet alle Ergebnisse den jeweiligen Projektbetreuern, bevor Angreifer sie ausnutzen können.

Obwohl JFrog immer wieder auf ähnlich geleakte Secrets stößt, war dieser Fall außergewöhnlich – denn die möglichen Folgen wären kaum zu überschätzen gewesen, wenn die Secrets in die falschen Hände geraten wären. Man hätte vermutlich Schadcode in alle PyPI-Pakete und sogar in die Programmiersprache Python selbst einschleusen können.

Das JFrog-Security-Forschungsteam identifizierte das geleakte Secret und meldete es umgehend dem Sicherheitsteam von PyPI, das den Token innerhalb von nur 17 Minuten widerrief. Auch wenn dieses Mal ein größeres Desaster verhindert werden konnte, macht der Vorfall eindringlich deutlich, welches verheerende Potenzial ein einziges geleaktes Secrets haben kann. Da PyPI zu den bedeutendsten Repositorys der Welt zählt, hätten die Auswirkungen weitreichend sein und Folgen für unzählige Nutzer und Projekte haben können. Dass ein solcher Vorfall selbst in einer so gut gewarteten und weit verbreiteten Infrastruktur wiePython/PyPI auftreten konnte, unterstreicht die Anfälligkeit aller Plattformen und Programmiersprachen und zeigt, dass diese Bedrohung jeden jederzeit treffen kann.

2025 JFrog Ltd. All rights reserved.

Wichtigste Erkenntnisse

Die Notwendigkeit von Datenredundanzen

Unternehmen und Security-Tools, die sich ausschließlich auf die Schwachstellendaten der NVD verlassen, laufen Gefahr, kritische CVE-Informationen zu übersehen – insbesondere aufgrund des erheblichen Bearbeitungsrückstands, den die NVD im vergangenen Jahr verzeichnete. Diese Verzögerungen führen dazu, dass neu entdeckte Schwachstellen und ihre potenziellen Auswirkungen möglicherweise nicht umgehend in die Datenbank aufgenommen werden. Dadurch können Unternehmen unwissentlich neuen Bedrohungen ausgesetzt sein. Um dieses Risiko zu minimieren, ist es unerlässlich, die NVD-Daten durch zusätzliche Quellen wie Hinweise der Anbieter und Threat-Intelligence-Feeds zu ergänzen, um zeitnah über kritische Schwachstellen informiert zu sein. Unternehmen sollten die Datenquellen ihrer Security-Scanning-Tools prüfen, um eine breite Abdeckung und Redundanz sicherzustellen.

Anwendbarkeit, Auswirkungen, Priorisierung

Angesichts der stetig wachsenden Anzahl von CVEs besteht die Gefahr, dass Sicherheits- und Development-Teams durch den Versuch, jede einzelne Schwachstelle zu bewerten, handlungsunfähig werden. Um Maßnahmen gezielt auf die Behebung von wirklich relevanten Schwachstellen auszurichten, ist das Verständnis der Anwendbarkeit, des Angriffsvektors und der potenziellen Auswirkung einer CVE in einer Anwendung entscheidend. Das JFrog-Security-Forschungsteam stößt weiterhin auf überhöhte Risikoeinstufungen in CVSS-Scores – ein Trend, der sich offenbar noch verstärkt, seit CISA als erster Authorized Data Publisher zur Erweiterung der CVE-Daten beiträgt.

Exposed Secrets können jeden treffen

Unternehmen müssen wachsam bleiben, um sich vor Exposed Secrets zu schützen, und diesen Schutz auf Entwickler ausweiten, die an persönlichen oder Gemeinschaftsprojekten arbeiten. Denn selbst wenn das System eines Developers im Zusammenhang mit einem privaten Projekt kompromittiert wird, können die Auswirkungen auch auf firmeneigene Systeme übergreifen. Es kann gravierende Folgen haben, wenn ein Exposed Secret oder ein geleaktes Token gefunden wird. Im Fall des von JFrog entdeckten Python/PyPI-Secrets hätte der Inhaber eines solchen Tokens Administratorzugriff auf sämtliche Repositorys von Python, PyPI und der Python Software Foundation gehabt. Das hätte einen groß angelegten Angriff auf die Software-Lieferkette ermöglichen können. Wenn es Python/PyPI treffen kann, kann es jeden treffen.

Raffinesse böswilliger Akteure

Böswillige Akteure werden in ihren Bemühungen, die Software-Lieferkette zu kompromittieren, immer kreativer und einfallsreicher. Im Fall der XZ Utils-Backdoor hat sich der Angreifer beispielsweise über mehrere Jahre hinweg einen glaubwürdigen Ruf als OSS-Entwickler aufgebaut und stark verschleierten Code verwendet, um selbst bei Code-Reviews unentdeckt zu bleiben. Andere Akteure nutzen KI-Tools aus, indem sie Instanzen identifizieren, in denen KI-Code-Assistenten „halluzinierte" Librarys vorschlagen. Dann erstellen sie schnell eine solche mit Schadcode versehene Library. Unternehmen müssen daher selbst gegenüber etablierten Open-Source-Projekten wachsam bleiben und klare Richtlinien für operationelle Risiken festlegen, um zu verhindern, dass solche "Overnight"-Bibliotheken unbeabsichtigt in die eigene Lieferkette gelangen.

2025 JFrog Ltd. All rights reserved.

Die Umsetzung von Sicherheitsmaßnahmen in Unternehmen nach heutigem Stand

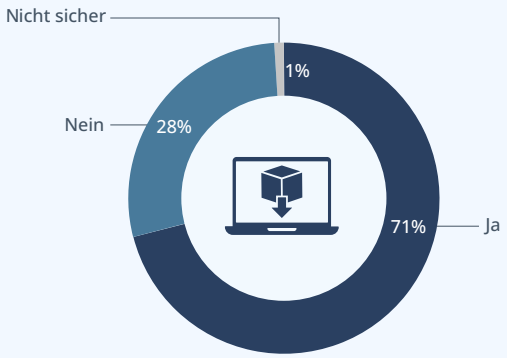
In diesem Jahr haben wir 1.402 Experten aus den Bereichen Security, DevOps und Engineering befragt, unseren Fragenkatalog erweitert und Erkenntnisse aus weiteren, von JFrog gesponserten Forschungsberichten einbezogen. Ziel war es, ein umfassenderes Bild davon zu gewinnen, wie Teams Anwendungsrisiken über den gesamten Softwareentwicklungs-Lebenszyklus (Software Development Lifecycle = SDLC) hinweg managen. Zwar verfügen die meisten Teams über Security-Frameworks und -Tools, jedoch waren wir überrascht, dass einige riskante Praktiken wie das direkte Herunterladen von Drittanbieter-Paketen oder -Librarys aus dem Internet weit verbreitet sind.

2025 JFrog Ltd. All rights reserved.

Sourcing-Beschränkungen

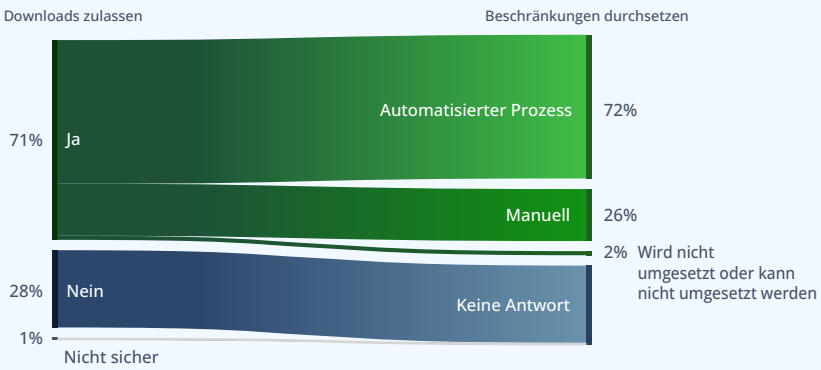
Eine der wirksamsten Maßnahmen, die Unternehmen ergreifen können, um Risiken in ihrer Software-Lieferkette zu minimieren, besteht darin, zu verhindern, dass diese überhaupt in die Lieferkette gelangen. Das geht über das “Shift-Left”-Prinzip hinaus, bei dem Sicherheitsmaßnahmen schon in die Entwicklungsphase integriert werden. Stattdessen erfordert es ein „Left of Left“-Vorgehen, bei dem Risiken bereits vor dem Eintritt in die Lieferkette blockiert werden.

Q **Erlaubt Ihr Unternehmen Entwicklern, Pakete oder andere Softwarekomponenten direkt aus öffentlichen Registrys oder anderen Quellen aus dem Internet herunterzuladen?** (Auftragsstudie, 2024)



Alarmierende 71 % der Unternehmen erlauben ihren Entwicklern, Softwarekomponenten direkt aus dem Internet herunterzuladen.Als Best Practice gilt: Entwickler sollten nicht die Möglichkeit haben, Pakete oder Librarys direkt aus dem Internet herunterzuladen – das Risiko ist einfach zu groß. Schon ein einziges kompromittiertes Entwicklergerät könnte ein gesamtes Unternehmen Angriffen aussetzen. Auch die Traceability ist gefährdet: Wenn direkte Downloads aus dem Internet erlaubt sind, lässt sich nicht nachvollziehen, welche Komponenten ein Entwickler tatsächlich herunterlädt.Dass dennoch so viele Unternehmen dieses Vorgehen zulassen, deutet auf einen konkreten Bedarf hin. Hier kann eine Artefaktmanagement-Lösung, die als Proxy für öffentliche Upstream-Registrys fungiert, besonders nützlich sein. Ein Artefakt-Repository mit Proxy-Funktionen dient als zentrale Kontrollinstanz für sämtliche Komponenten, die in die Software-Lieferkette gelangen, und ermöglicht so, diese zu tracken und zu sichern. Mit einer solchen Lösung können Unternehmen das Risiko dieses Vorgehens verringern und potenziell schwerwiegenden Schaden verhindern.

Q **Wie trackt und erwirkt Ihr Unternehmen die Sourcing-Beschränkungen für Pakete oder andere Softwarekomponenten, die direkt aus öffentlichen Registrys oder anderen Quellen aus dem Internet stammen?** (Auftragsstudie, 2024)



Mehr als jeder vierte Befragte (26 %) gibt an, dass sein Unternehmen die Sourcing-Beschränkungen für Pakete oder andere Softwarekomponenten, die direkt aus öffentlichen Registrys oder anderen Internetquellen stammen, manuell trackt und durchsetzt.

2025 JFrog Ltd. All rights reserved.

Mit 72 % ist der Anteil der Befragten, die angeben, automatisierte Prozesse einzusetzen, unerwartet hoch. Allerdings könnte dieses Ergebnis davon beeinflusst sein, auf welche Phase des SDLC sich die Befragten beziehen. So ist es beispielsweise denkbar, dass Developer Pakete und Abhängigkeiten zunächst manuell prüfen, bevor sie in die Codebasis übernommen werden, während automatisierte Prozesse erst im CI/CD-Zyklus oder später bei Audits von Release-Builds ausgeführt werden. Die Herausforderung bei dieser Herangehensweise besteht darin, dass sie zusätzlichen Aufwand seitens der Entwickler erfordert und, wie an anderer Stelle in diesem Report bereits erwähnt, früh im SDLC zu Exposures führen kann.

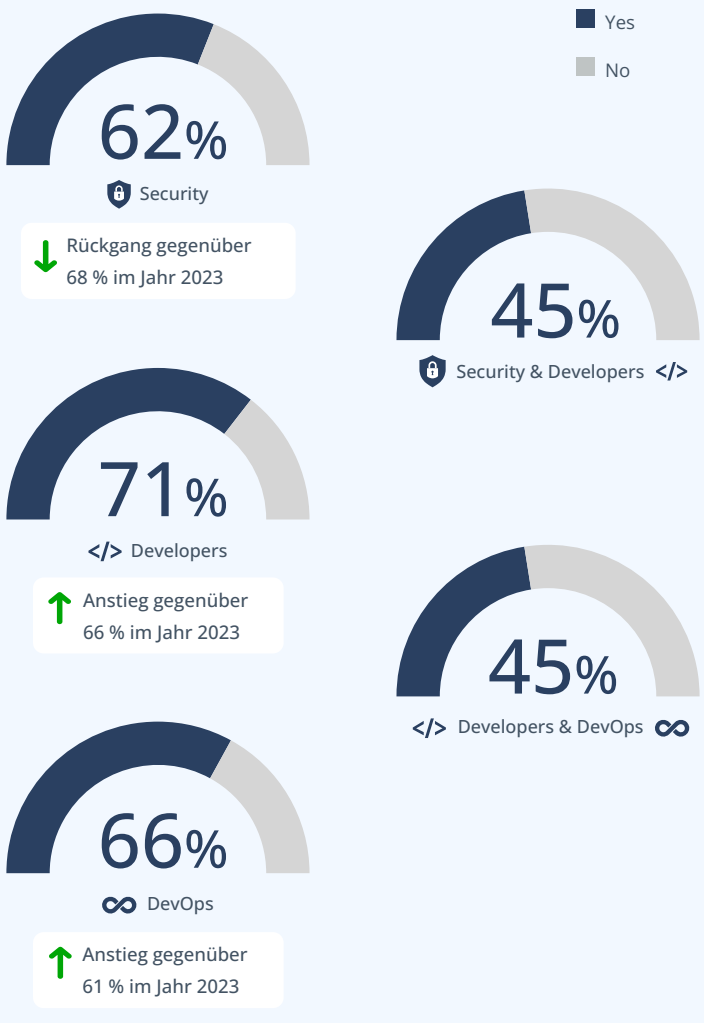
19 % der befragten Unternehmen geben an, dass Entwickler Pakete direkt aus dem Internet herunterladen dürfen und dabei manuelle Verfahren zur Durchsetzung von Sourcing-Beschränkungen genutzt werden. Das erfordert einen erheblichen manuellen Aufwand und stellt keinen effektiven Ansatz zur Blockierung potenzieller Risiken dar.

52 % der Befragten geben an, dass sie Entwicklern ebenfalls erlauben, Komponenten direkt aus dem Internet herunterzuladen, jedoch gleichzeitig automatisierte Prozesse einsetzen, um Sourcing-Beschränkungen zu tracken und durchzusetzen.

Wer ist dafür zuständig, Softwarepakete, Librarys und Frameworks auf dem neuesten Stand zu halten: Security oder Entwickler? Wählen Sie alle zutreffenden Antworten aus. (Auftragsstudie, 2024)

Im Vergleich zu den Vorjahren zeigt sich, dass Entwickler mittlerweile eine aktivere Rolle bei der Verwaltung der neuesten Pakete übernehmen. Dennoch liegt die Verantwortung weiterhin auch bei Security-Experten, insbesondere wenn es um die Prüfung und Freigabe zum Einsatz von Paketen geht. In Unternehmen sind in der Regel mehrere Teams für den Akquisitionsprozess verantwortlich, wobei die Kombinationen „Entwickler + Security“ und „Entwickler + DevOps“ am häufigsten genannt werden.

Um die Entwicklungsgeschwindigkeit zu erhöhen, müssen Unternehmen Ansätze und Lösungen implementieren, die Entwickler in die Lage versetzen, neue und aktuelle Versionen von Librarys eigenständig einzubinden und die Freigabe jener Pakete zu automatisieren, die den Sicherheitsrichtlinien entsprechen. Zudem sollte ein Waiver-Management-Programm etabliert werden, das dem zuständigen Team – sei es AppSec oder Security – ermöglicht, dieses Waiver-Programm in die übergeordnete Risikomanagement-Strategie zu integrieren.

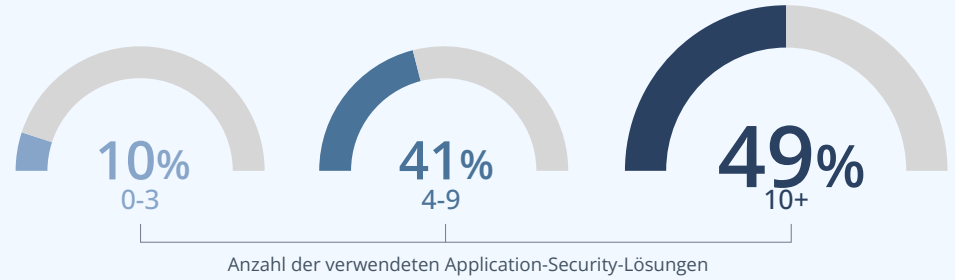


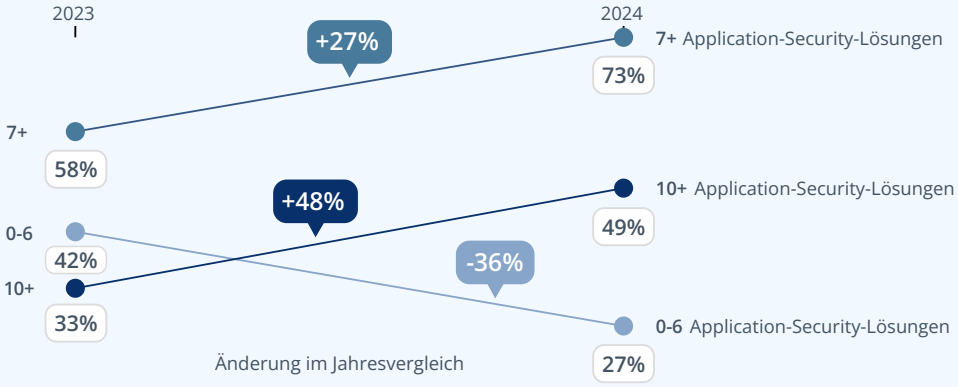
2025 JFrog Ltd. All rights reserved.

Scanning, scanning, scanning

Obwohl Unternehmen mehr Sicherheitstools einsetzen als je zuvor, bleiben Lücken bestehen. Fehlendes Scannen sowohl von Code als auch von Binärdateien sowie uneinheitliches Scannen im SDLC und in der Produktion zählen zu den häufigsten blinden Flecken.

Wie viele Application-Security-Lösungen verwenden Sie?
(Auftragsstudie, 2023 & 2024)





Die Befragten geben an, im Jahr 2024 mehr Anwendungssicherheit-slösungen einzusetzen als 2023. Ende 2024 nutzten 73 % sieben oder mehr Application-Security-Lösungen – im Vergleich zu 58 % im Vorjahr.

Das steht im Widerspruch zu den Erwartungen, die man angesicht des Marktfokus auf Tool-Konsolidierung und den Rückmeldungen von JFrog-Kunden in Führungspositonen über ihren Wunsch nach einem schlankeren, sicheren Softwareentwicklungsprozess haben könnte. Die Daten deuten darauf hin, dass ASPM (Application Security Posture Management) – eine neue Kategorie von Tools, die es Unternehmen ermöglicht, mehrere Scanning-Lösungen parallel zu nutzen und dabei doppelte Ergebnisse herauszufiltern – von Unternehmen eingesetzt wird, um eine Überabdeckung aufrechtzuerhalten und so das Risiko zu minimieren, etwas zu übersehen. Allerdings ist ASPM lediglich eine Notlösung für den Wildwuchs an Sicherheitstools und keine nachhaltige Maßnahme. Wir gehen nicht davon aus, dass das Wachstum bei der Anzahl eingesetzter Sicherheitstools im nächsten Jahr anhält, da Unternehmen ihre Konsolidierungsbemühungen neu ausrichten werden.

2025 JFrog Ltd. All rights reserved.

Setzt Ihr Unternehmen Sicherheitsscans auf der Code- oder Binärebene ein? (Auftragsstudie, 2024)

Code- und Binärscans

43%

Rückgang gegenüber 56 % im Jahr 2023

Nur Code-Scans

29%

Anstieg gegenüber 27 % im Jahr 2023

Nur Binärscans

25%

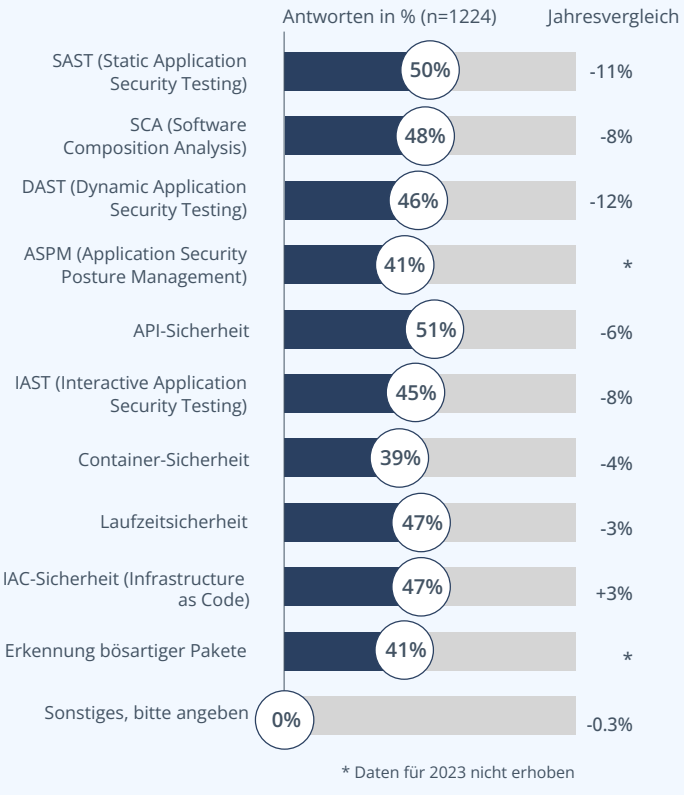
Anstieg gegenüber 12 % im Jahr 2023

Die Beliebtheit von Sicherheitsscans auf Binärebene hat sich in diesem Jahr verdoppelt: 25 % der Befragten geben an, Sicherheitsmaßnahmen auf dieser Ebene umzusetzen – gegenüber nur 12 % im Jahr 2023.

43 % der Befragten geben an, dass ihr Unternehmen Sicherheitsscans sowohl auf der Code- als auch auf Binärebene durchführt – ein leichter Rückgang gegenüber 56 % im Jahr 2023. Dieser Trend ist durchaus besorgniserregend, denn idealerweise sollten Sicherheitsscans sowohl auf Code- als auch auf Binärebene erfolgen, um Risiken so früh wie möglich zu erkennen und zu vermeiden. Ein Grund für dieses Vorgehen ist, dass bestimmte Arten von Schwachstellen sich ausschließlich auf Binärebene manifestieren.

So können beispielsweise in Binärdateien eingeschleuste Secrets oder durch den Compiler eingefügte Speicherfehler Sicherheitsprobleme verursachen, die im Quellcode nicht vorhanden sind oder versehentlich in Builds verbleiben, die schließlich in die Produktion gelangen.

Welche Arten von Application-Security-Lösungen setzen Sie ein? (Auftragsstudie, 2024)



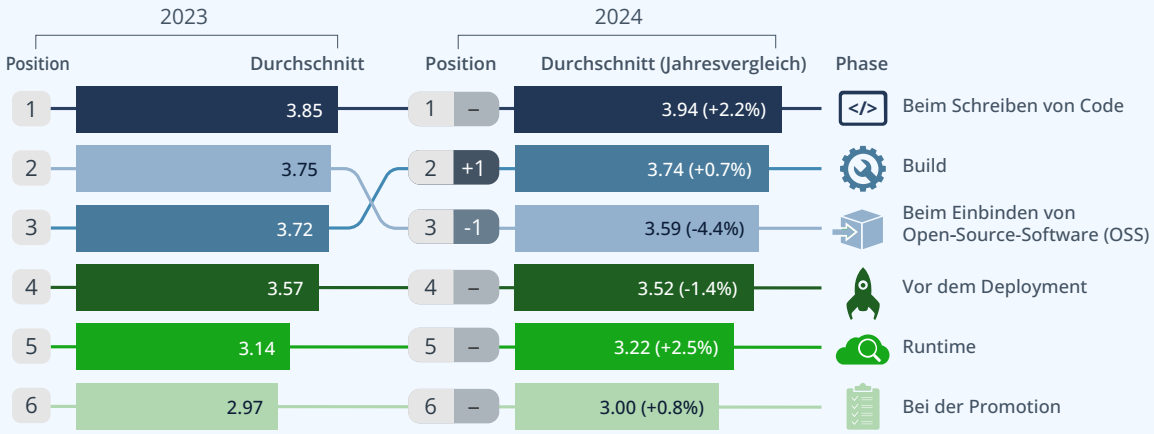
* Daten für 2023 nicht erhoben

Es gibt kein Tool, das von einer überwältigenden Mehrheit verwendet wird. Nur API-Sicherheit und SAST erreichen oder übertreffen die 50-Prozent-Marke. Angesichts des anhaltenden Fokus auf „Shift-Left"-Strategien ist die weite Verbreitung von SAST nachvollziehbar. Ebenso wenig überrascht es, dass Unternehmen in API-Security-Tools investieren – angesichts der Prävalenz moderner Microservice-Anwendungen, bei denen APIs eine potenzielle Schwachstelle für Angriffe durch Bedrohungsakteure darstellen.

Relativ zueinander bleiben die Nutzungsraten der verschiedenen Tooltypen im Jahresvergleich stabil, allerdings ist der Gesamtprozentsatz der Befragten, die einen bestimmten Tooltyp verwenden, zurückgegangen. Das ist besonders bemerkenswert, wenn man bedenkt, dass – wie weiter oben in diesem Report angeführt – gleichzeitig die Gesamtzahl der Sicherheitstools steigt. Das könnte darauf hindeuten, dass es Überschneidungen bei den verwendeten Tooltypen gibt oder dass verschiedene Teams jeweils eigene bevorzugten Sicherheitstools nutzen, die dieselbe Funktionalität bieten wie die bevorzugten Tools anderer Teams. Unternehmen sollten ein Audit ihrer Sicherheitstools in Betracht ziehen, um Redundanzen und Lücken in ihrer Sicherheitslandschaft zu identifizieren.

2025 JFrog Ltd. All rights reserved.

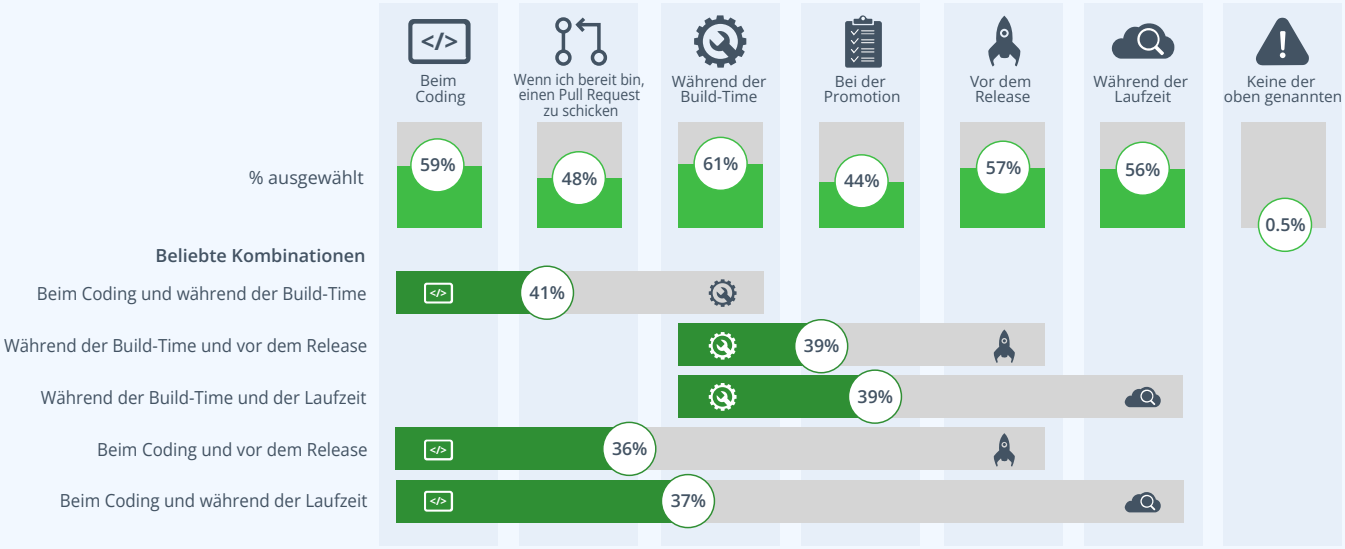
In welcher Phase des SDLCs ist Ihrer Meinung nach der beste Zeitpunkt für Sicherheitsmaßnahmen? (Auftragsstudie, 2024)



Auf die Frage, an welcher Stelle im Softwareentwicklung-Lebenszyklus Sicherheitsmaßnahmen am sinnvollsten sind, blieben die drei am häufigsten genannten Punkte im Vergleich zum Vorjahr unverändert:

Beim Schreiben von Code	Build	Beim Einbinden von Open-Source-Software (OSS)
M=3.94	M=3.74	M=3.59

An welchem Punkt im Development führt Ihr Unternehmen in der Regel Sicherheitsscans durch? Wählen Sie alles Zutreffende aus. (Auftragsstudie, 2024)



„Beim Coding" ist nach wie vor der am häufigsten genannte Punkt innerhalb des SDLC, an dem Unternehmen typischerweise Sicherheitsscans durchführen. Fast drei von fünf Befragten geben an, dass ihr Unternehmen in dieser Phase Sicherheitsscans durchführt.

41%	Beim Coding	+	Während der Build-Time
39%	Während der Build-Time	+	Vor dem Release
30%	Während der Build-Time	+	Während der Laufzeit
36%	Beim Coding	+	Vor dem Release
37%	Beim Coding	+	Während der Laufzeit

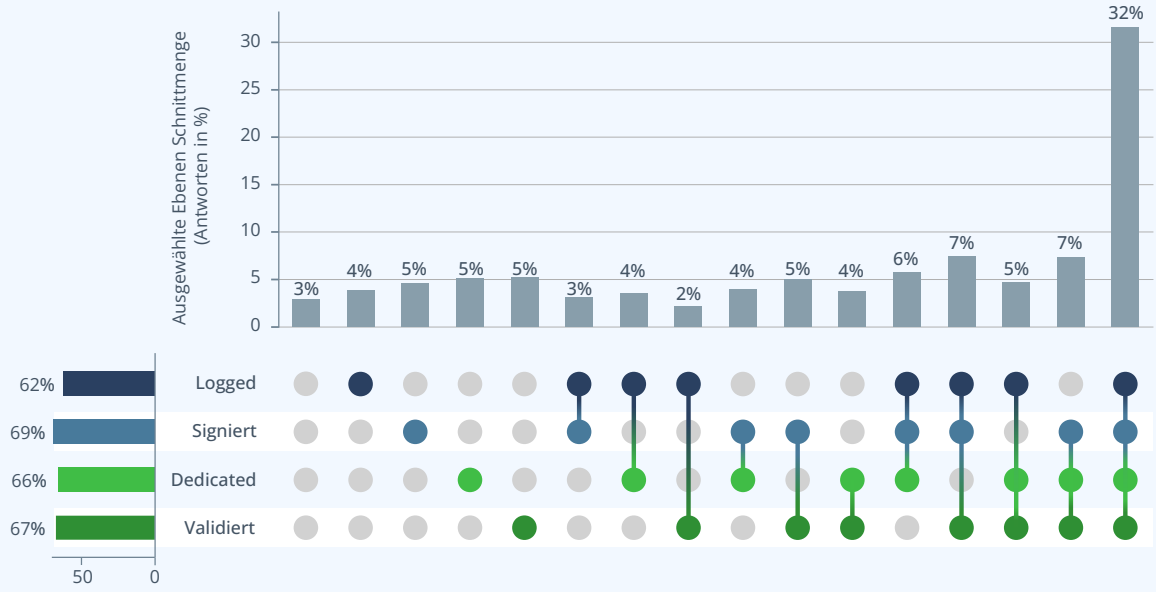
2025 JFrog Ltd. All rights reserved.

Transparenz und Kontrolle über die gesamte Anwendungspipeline schaffen

Es erscheint naheliegend, dass Unternehmen, die Anwendungen entwickeln, Risiken auch ganzheitlich auf Anwendungsebene managen. Doch obwohl viele Unternehmen Anwendungen bereits über den gesamten SDLC hinweg definieren und tracken, unterscheidet sich das Maß an Kontrolle und Rückverfolgbarkeit erheblich. Um ein sicheres Risikomanagement und Vertrauen in die freigegebene Software zu gewährleisten, müssen diese beiden Elemente unbedingt gestärkt werden.

Kontrolle beginnt bereits in der Sourcing-Phase, noch bevor überhaupt eine „Anwendung" existiert. Die in den Entwicklungsprozess integrierten Komponenten und Librarys prägen den Sicherheitsstatus des Endprodukts maßgeblich. Durch eine sorgfältige Bewertung und Auswahl von Drittanbieter-Ressourcen können Unternehmen Risiken minimieren, noch bevor diese sich in der Anwendung manifestieren.

Welche der folgenden Ebenen des Security-Frameworks sind in Ihrem Unternehmen implementiert? (Auftragsstudie, 2024)



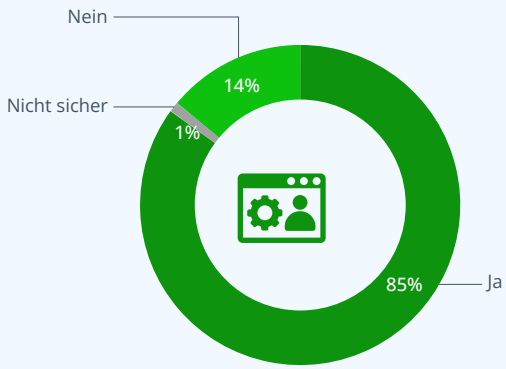
- Logged – Paket-Signaturen werden vor der Veröffentlichung validiert
- Signiert – Pakete werden auf einem dedizierten Host erstellt
- Dedicated – Build-Daten und Metadaten der Pakete sind signiert
- Validiert – Build-Metadaten der Pakete werden protokolliert

Die Mehrheit der befragten Unternehmen setzt auf Frameworks wie Supply-Chain Levels for Software Artifacts (SLSA), um die Sicherheit und Integrität von Software-Lieferketten zu verbessern. Die Daten zeigen, dass ein Großteil zumindest ein SLSA-Level implementiert hat, während etwas mehr als ein Drittel der Befragten alle SLSA-Stufen umsetzt.

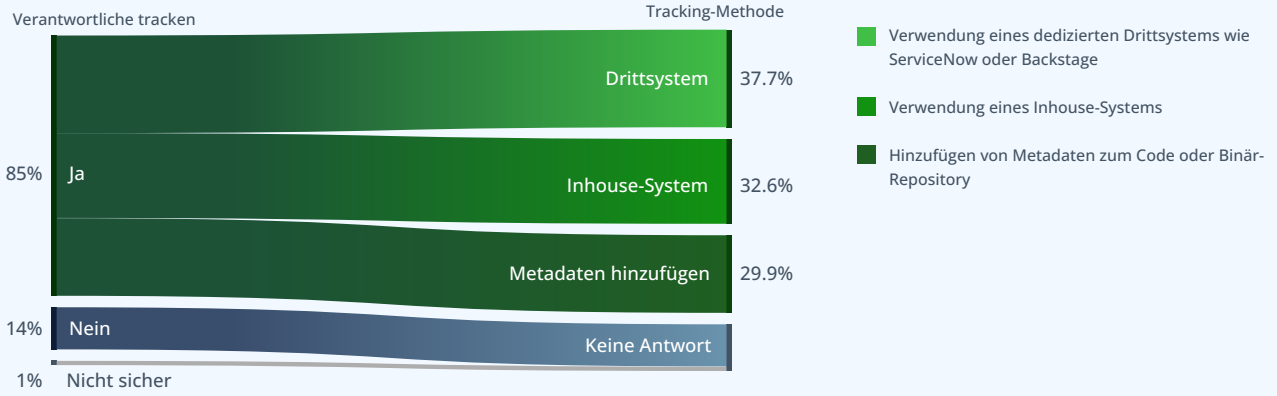
2025 JFrog Ltd. All rights reserved.

Den Eigentümer jeder Anwendung tracken

Erfassen Sie für jede Anwendung, die in Ihrem Unternehmen entwickelt wird, den Eigentümer der Anwendung (d. h. das Team oder die Einzelperson)?
(Auftragsstudie, 2024)



Wie behalten Sie bei jeder Anwendung, die in Ihrem Unternehmen entwickelt wird, den Überblick über deren Eigentümer? (Auftragsstudie, 2024)

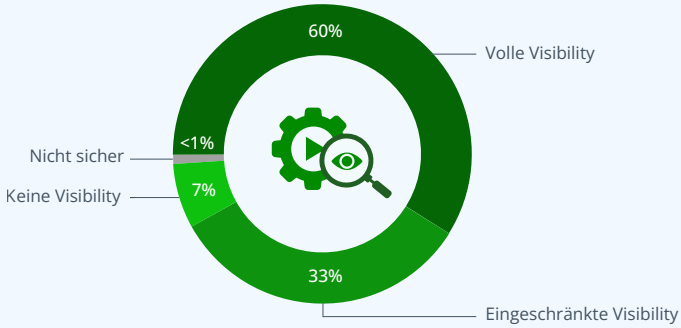


Das Tracking der Eigentümer von Anwendungen und der dazugehörigen Microservices ist aus vielen Gründen entscheidend. Beispielsweise um Probleme schnell zu beheben, Abhängigkeiten zwischen Anwendungen zu verstehen und ordnungsgemäße Governance- und Business-Continuity-Pläne umzusetzen. Zwar tracken die meisten Unternehmen die Eigentümer der entwickelten Anwendungen, doch die Art und Weise, wie sie das tun, variiert stark. Die Antworten verteilen sich dabei nahezu gleichmäßig auf dedizierte Drittsysteme, hauseigene Systeme und die Verwendung von Metadaten im Code oder Binär-Repository. Kein einziger Befragter gibt an, eine andere Methode als diese drei zu verwenden.

2025 JFrog Ltd. All rights reserved.

Haben Sie Einblick in die Herkunft der in der Produktionsumgebung ausgeführten Software (d. h. wer den Code für einen bestimmten Service committet hat, welche Tests und Validierungen durchgeführt wurden und woher die Abhängigkeiten stammen)? (Auftragsstudie, 2023 & 2024)

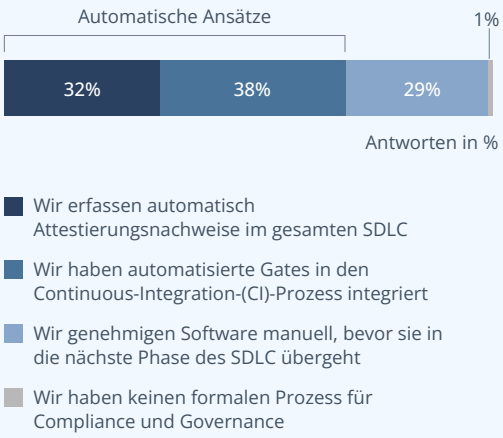
Nur 60 % der Unternehmen geben an, vollständigen Überblick über die Herkunft der in der Produktionsumgebung ausgeführten Software zu haben. Etwa ein Drittel (33 %) verfügt lediglich über begrenzten Einblick und erfreulicherweise haben weniger als 8 % keinen Einblick oder sind sich über die Herkunft ihrer Software nicht im Klaren.



Einblick in die Herkunft von Software ist entscheidend, um die Qualität und Sicherheit veröffentlichter Software zu garantieren – und wird zunehmend zu einer obligatorischen Anforderung verschiedener staatlicher Regulierungen. Die rund 8 %, die angeben, keinen Einblick in die Herkunft ihrer Software zu haben, sollten zumindest eine Bestandsaufnahme ihrer Codebasen und aller externen Pakete vornehmen und sicherstellen, dass eine automatisierte CI/CD-Pipeline implementiert ist, die Build-Versionen trackt und zuweist. Auch wenn viele die Versionskontrolle als selbstverständlich ansehen, ist es möglich, dass ein Teil dieser knapp 8 % noch keine robuste Lösung zur Versionskontrolle implementiert hat, um Codeänderungen während der Entwicklung zu tracken.

Wie stellen Sie sicher, dass die Standards für Softwaretests und -qualität während der Entwicklung und dem Freigabeprozess im Hinblick auf Compliance und Governance eingehalten werden? (Auftragsstudie, 2023 & 2024)

Auch die Methoden zur Sicherstellung von Test- und Qualitätsstandards variieren von Unternehmen zu Unternehmen. Während die Mehrheit (70 %) auf automatisierte Ansätze setzt, nutzt fast ein Drittel (29 %) immer noch manuelle Genehmigungen, um die verschiedenen Phasen des SDLC zu durchlaufen.



2025 JFrog Ltd. All rights reserved.

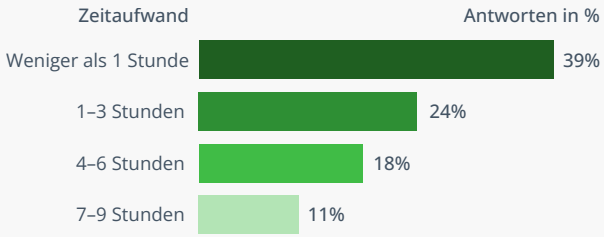
Wie viel Zeit kosten Ihrem Unternehmen die Sicherheitsmaßnahmen?

In einer von IDC durchgeführten und von JFrog in Auftrag gegebenen Studie geben 60 % der Fachleute an, dass ihr Entwicklungs- und/oder Security-Team in der Regel vier oder mehr Tage pro Monat damit verbringt, Anwendungsschwachstellen zu beheben. Die dadurch entstehenden Kosten belaufen sich im Durchschnitt auf etwa 28.000 US-Dollar pro Entwickler und Jahr. Das hat nicht nur finanzielle Auswirkungen, sondern wirkt sich auch negativ auf die Developer Experience (DevEx) aus.

IDC: Die wahren Kosten von DevSecOps
Veröffentlicht: September 2024 | IDC #US52537524

Entwicklerzeit außerhalb der regulären Arbeitszeit zur Behebung von Sicherheitsproblemen

Developer verbringen durchschnittlich rund 3,6 Stunden pro Woche außerhalb der Arbeitszeit mit der Behebung von Sicherheitsproblemen. Das schafft ein Umfeld, das Burnout begünstigt.



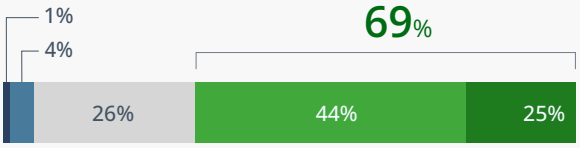
~3.6
Stunden pro Woche verbringen Entwickler außerhalb ihrer regulären Arbeitszeit mit der Behebung von Sicherheitsproblemen

Ausgaben für sicherheitsrelevante Aktivitäten

Die meisten Unternehmen geben jährlich mehr als 28.100 US-Dollar pro Entwickler für sicherheitsrelevante Aufgaben aus. Auch wenn DevSecOps eine wirtschaftliche Notwendigkeit und unerlässlich für die Entwicklung sicherer Anwendungen ist, verschwenden ineffiziente oder schlecht implementierte Tools und Prozesse die Zeit der Entwickler und verursachen erhebliche Betriebskosten.

$28K
werden jährlich pro Entwickler für sicherheitsrelevante Aufgaben aufgewendet

Zu viel Kontextwechsel



- Stimme voll und ganz zu
  - Ich wechsle ständig zwischen Tools oder Umgebungen
- Stimme zu
  - Ich wechsle häufig zwischen Tools oder Umgebungen
- Unentschlossen
- Stimme nicht zu
  - Ich wechsle gelegentlich zwischen Tools oder Umgebungen
- Stimme überhaupt nicht zu
  - Ich wechsle selten oder nie zwischen Tools oder Umgebungen

69 % der Entwickler stimmen zu, dass ihre sicherheitsrelevanten Tätigkeiten einen häufigen Kontextwechsel erfordern. Unternehmen, die die DevEx verbessern möchten, sollten berücksichtigen, dass ein regelmäßiger Tool-Wechsel diesen Bemühungen schadet und die Bereitschaft von Entwicklern verringert, sich mit Sicherheitsmaßnahmen auseinanderzusetzen.

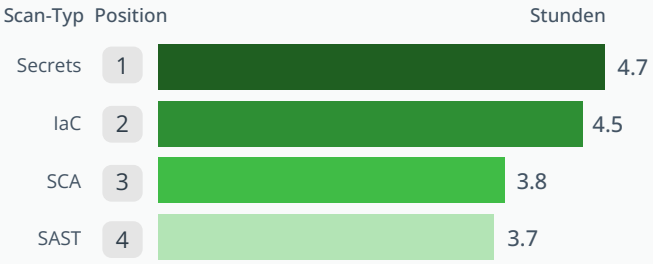
69 %
der Entwickler stimmen zu , dass ihre sicherheitsrelevanten Tätigkeiten einen häufigen Kontextwechsel erfordern.

2025 JFrog Ltd. All rights reserved.

Zeitaufwand nach Scan-Typ

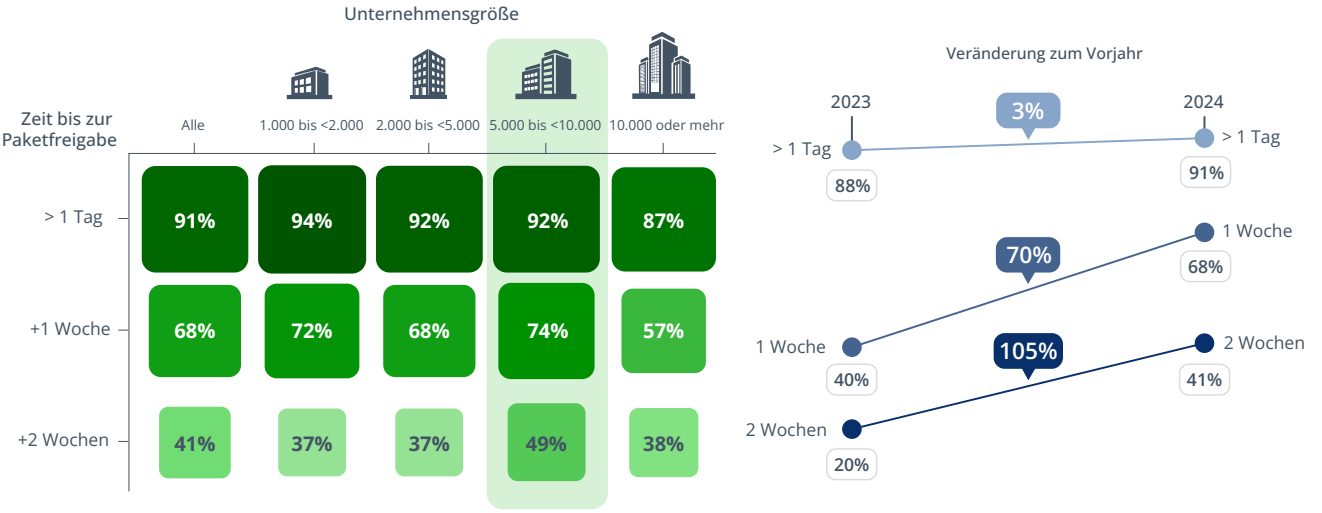
Entwickler verbringen den Großteil ihrer Zeit mit dem Scannen auf Secrets, was entweder auf einen Schulungsbedarf zu Programmierpraktiken im Umgang mit Token und Secrets oder auf den Bedarf an effizienteren Tools zur Handhabung von Secrets hinweist. Zudem ist entscheidend, sicherzustellen, dass keine Secrets im Code verbleiben. Das wäre vergleichbar damit, den Hausschlüssel samt Adressangabe auf der Straße zu verlieren. Oft ermöglichen im Code hinterlassene Secrets Angreifern ungehinderten Zugriff auf kritische Systeme und Daten.

4.7
Stunden verbringen Entwickler mit dem Scannen auf Secrets



Die im Juni 2024 online durchgeführte IDC-Studie erfasste Antworten von 210 Entwicklern, Leitern von Development-Teams, Managern und Product Ownern aus den USA und Europa, die DevSecOps nutzen. Ziel der Studie war es, Informationen über den Business Impact des Zeitaufwands von Entwicklern für DevSecOps, DevSecOps-Tools und -Aufgaben, die Entwicklerzeit beanspruchen, den Stellenwert des Zeitaufwands von Entwicklern für DevSecOps sowie die Auswirkungen von Security-Tasks auf den Workflow und die Zufriedenheit von Developern zu ermitteln.

Wie lange dauert es in der Regel, bis die Freigabe zur Nutzung eines neuen Pakets bzw. einer neuen Library erfolgt? (Auftragsstudie, 2023 & 2024)



Entwickler warten länger denn je auf neue Pakete. Mittelgroße Unternehmen mit 5.000 bis 10.000 Mitarbeitern warten am längsten: 92 % warten länger als einen Tag, 74 % länger als eine Woche und 49 % sogar zwei Wochen oder mehr. Obwohl es eine positive Entwicklung ist, dass Developer stärker in den Prozess eingebunden sind, bleibt dieser nach wie vor ineffizient. Das liegt wahrscheinlich an den Reviews und anderen manuellen Prozessen. Es braucht mehr, um die Integration neuer Komponenten als Selfservice-Prozess zu etablieren.

2025 JFrog Ltd. All rights reserved.

Wichtigste Erkenntnisse

71%

Fehlende grundlegende Praktiken zur Sicherung der Software-Lieferkette

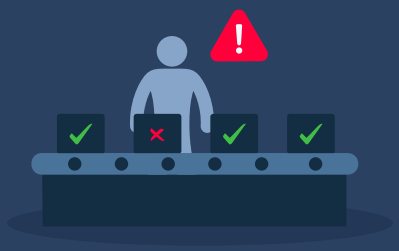
Unternehmen müssen kontrollieren oder zumindest einen guten Überblick darüber haben, was über ihre Entwickler und die in Anwendungen referenzierten Abhängigkeiten in ihre Software-Lieferkette gelangt. Dass über 71 % der Unternehmen ihren Entwicklern erlauben, Pakete direkt aus dem Internet herunterzuladen, ist bedenklich und stellt einen eklatanten Verstoß gegen die Best Practices der Software-Lieferketten-Sicherheit dar. Jedes Unternehmen sollte über eine Artefakt-Management-Lösung verfügen, die öffentliche Registrys über einen Proxy anbindet.

Mehr Scanning-Tools, mehr Probleme?

Unternehmen setzen zwar mehr Sicherheitstools ein als je zuvor – doch wirkt sich das positiv oder negativ auf ihren Sicherheitsstatus aus? Unabhängig davon bestehen weiterhin Lücken in der Abdeckung und viele Unternehmen scannen nicht sowohl auf Code- als auch auf Binärebene – was ein ernstzunehmendes Problem darstellt.

DevSecOps ohne Einbußen bei der DevEx

Sicherheitsbemühungen kosten Entwickler jede Woche mehrere Stunden Zeit. Unternehmen können und sollten nach Möglichkeiten suchen, den Einfluss von Sicherheitsaufgaben auf Entwickler zu reduzieren, ohne dabei den Sicherheitsstatus ihrer Anwendungen zu gefährden. Der Schlüssel liegt in intelligenter Priorisierung, kontextualisierten Ergebnissen und Automatisierung.



Aufwertung des Anwendungsmanagements

85 % der Unternehmen tracken die Eigentümer von intern entwickelten Anwendungen, doch die Art und Weise, wie sie die Anwendungsstandards sicherstellen, variiert erheblich: Fast ein Drittel setzt dabei auf manuelle Prozesse, um ihre Software von einer Phase in die nächste zu überführen. Jede manuelle Intervention birgt ein Risiko – sei es unbeabsichtigt oder absichtlich – und zeigt auf, wo deutliches Optimierungspotenzial besteht.

2025 JFrog Ltd. All rights reserved.

Die nächste Risikodimension: Entwicklung von KI und Machine Learning

Nahezu jedes Sicherheits-Tool und zunehmend auch Developer-Tools werben mittlerweile mit KI-gestützten Funktionen, um die Entwicklung zu beschleunigen und die Erkennung sowie Behebung von Schwachstellen zu verbessern. In diesem Abschnitt des Reports liegt der Fokus jedoch nicht auf dem Einsatz von KI, sondern auf der Entwicklung von KI-Tools.

Die Software-Lieferkette für künstliche Intelligenz und maschinelles Lernen (KI/ML) stellt die nächste Risikodimension für Unternehmen dar und befindet sich auf der Reifekurve deutlich weiter “links” als die klassische Softwareentwicklung. Laut einer von JFrog beauftragten und in Zusammenarbeit mit InformationWeek durchgeführten Studie geben 79 % der Unternehmen an, dass Sicherheitsbedenken die Nutzung und/oder Integration von KI/ML-Funktionen in Software verlangsamen.

2025 JFrog Ltd. All rights reserved.

Trends bei der Einführung von KI und DevSecOps

Die von InformationWeek durchgeführte Umfrage untersuchte, wie gut Softwareentwickler und Cybersecurity-Teams die Bedeutung der Integration von Anwendungssicherheit in den Softwareentwicklungs-Lebenszyklus verstehen. Außerdem wurde analysiert, wie Teams ihr Unternehmen vor bösartigem Code schützen und die unsachgemäße Verwendung von KI-Technologien vermeiden. Zu den wichtigsten Erkenntnissen der Umfrage gehören:

KI-Integration und DevSecOps: Immer einen Schritt voraus, immer auf der sicheren Seite

Veröffentlicht: September 2024 | InformationWeek & JFrog

Mangelndes Vertrauen in die Sicherheit von KI-Technologien auf Unternehmensebene

79 % der Unternehmen geben an, dass Sicherheitsbedenken die Nutzung und/oder Integration von KI/ML-Features in Software hemmen

Die Top 3 Sicherheitsbedenken im Zusammenhang mit KI innerhalb von Unternehmen sind Datenexposition durch die Nutzung von LLMs, schädlicher Code in KI-Modellen und KI-Bias

64 % der Unternehmen sind entweder überhaupt nicht oder nur teilweise zuversichtlich, dass sie neue und bevorstehende Vorschriften zur Nutzung von KI in Software einhalten können

Die Transparenz der KI-Lieferkette ist mangelhaft

49 % der Unternehmen haben keine verlässliche Möglichkeit, die Nutzung von ML-Modellen in ihren Apps zu kontrollieren

Weniger als 1/4 der Unternehmen verfügt über eine "Single Source of Truth" für alle Softwarekomponenten, einschließlich KI-Modellen

Mehr als 2/3 der Unternehmen haben keine zuverlässige Methode zum Tracking von Open-Source-Paketen in ihrer Software, die transitive Abhängigkeiten zu ML-Modellen enthalten

Unzureichende Richtlinien für den Einsatz von KI

58 % der Unternehmen haben entweder keine Richtlinien oder wissen nicht, ob sie Richtlinien haben, die Regeln für die Nutzung von Open-Source-KI-Modellen oder -Komponenten durch Entwickler festlegen

60 % der Unternehmen verfügen über keine Richtlinien dafür, wie Entwickler ihre Trainingsdaten beziehen oder lizenzieren

Die Umsetzung ist noch unzuverlässiger

68 % der Befragten geben an, dass sie keine Möglichkeit haben, die Nutzung von KI-Komponenten durchzusetzen, oder dass sie sich dabei auf eine manuelle Überprüfung verlassen müssen

59 % geben an, dass sie über keine vernünftigen Prozesse verfügen oder auf eine manuelle Überprüfung angewiesen sind, um Richtlinien für Trainingsdaten umzusetzen

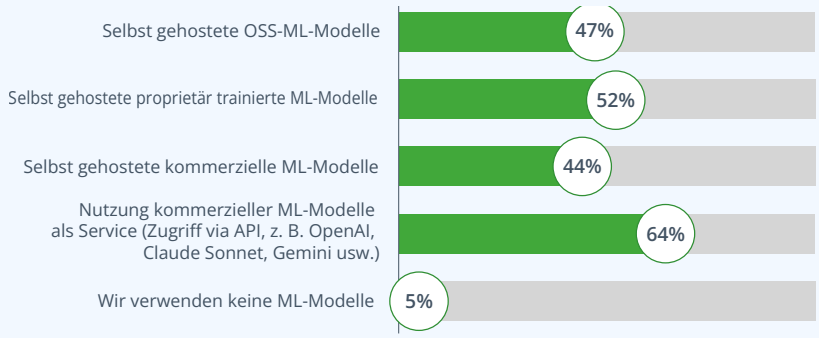
Die Studie, mit der JFrog InformationWeek beauftragte, wurde im Mai 2024 online durchgeführt und erfasste die Antworten von 210 überwiegend in Nordamerika ansässigen IT- und Cybersecurity-Experten. Die Befragten kamen aus Unternehmen aller Größenordnungen und mit Positionen von der Führungsebene bis hin zu Personal. Mehr als 21 Branchen sind vertreten, darunter Consulting, Bank- und Finanzdienstleistungen, Bildung, Behörden, Technologie, Gesundheitswesen und Industrie.

Auch wenn die InformationWeek-Studie interessante Trends aufgedeckt hat, geht der restliche Teil dieses Abschnitts näher darauf ein, wie Unternehmen KI-Services konkret in ihre Anwendungen integrieren und deren Nutzung regeln.

2025 JFrog Ltd. All rights reserved.

Nutzung, Governance und Scanning von ML-Modell-Artefakten

Welche Methode nutzen Sie primär, um ML-Modelle in die von Ihnen entwickelten Anwendungen zu integrieren? (Auftragsstudie, 2024)

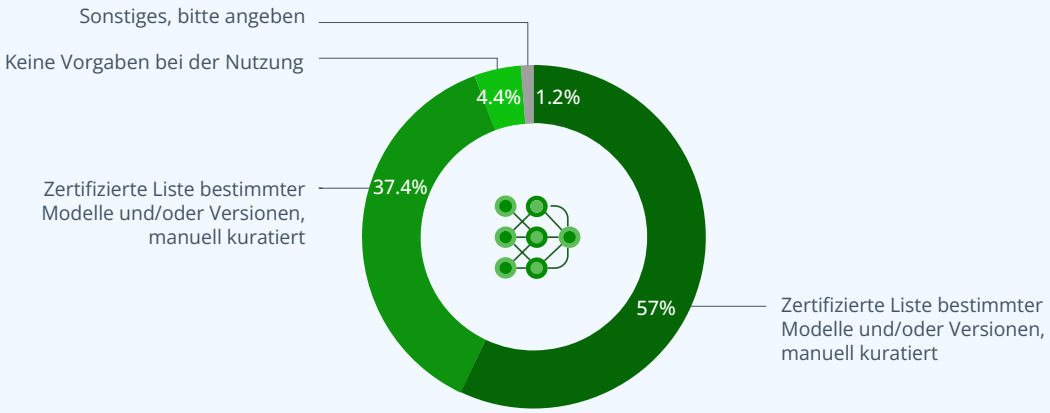


Die Art und Weise, wie Unternehmen KI-Services und -Anwendungen realisieren, ist unterschiedlich und die Untersuchung zeigt, dass Unternehmen mehrere Methoden parallel nutzen.

Mit Abstand am beliebtesten ist der Einsatz kommerzieller Modelle über APIs (64 %). Dieser Ansatz ermöglicht es, schnell und ohne Vorlaufkosten für Entwicklung oder Infrastruktur auf leistungsstarke, universell einsetzbare KI-Funktionen zuzugreifen.

Allerdings ist auch zu sehen, dass Unternehmen zunehmend in selbst gehostete Modelle investieren. Mehr als die Hälfte entwickelt selbst gehostete proprietäre Modelle für ihre spezifischen Geschäftsanforderungen. Fast jeder zweite Befragte gibt an, dass selbst gehostete OSS-Modelle die primäre Methode zur Nutzung von Machine-Learning-Modellen darstellen.

Wie steuern Sie die Nutzung von ML-Modell-Artefakten in Ihrem Entwicklungsteam? (Auftragsstudie, 2024)



Mehr als jede Dritte Fachkraft (37 %) gibt an, die Nutzung von ML-Modell-Artefakten über eine manuell kuratierte Liste bestimmter Modelle und/oder Versionen zu steuern.

Wie die InformationWeek-Studie zeigt, verfügen 49 % der Unternehmen über keine verlässliche Methode, um die Nutzung von ML-Modellen in ihren Anwendungen zu kontrollieren. Das könnte erklären, warum 4 % der Befragten bewusst keinerlei Maßnahmen ergreifen – weder manuelle noch anderweitig –, um zu kontrollieren, welche Modelle Entwickler verwenden.

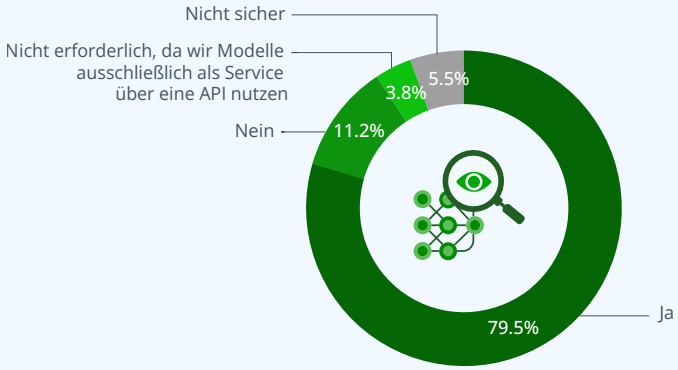
16 % aller Befragten nutzen selbst gehostete OSS-Modelle und regeln die Verwendung von Modell-Artefakten manuell.

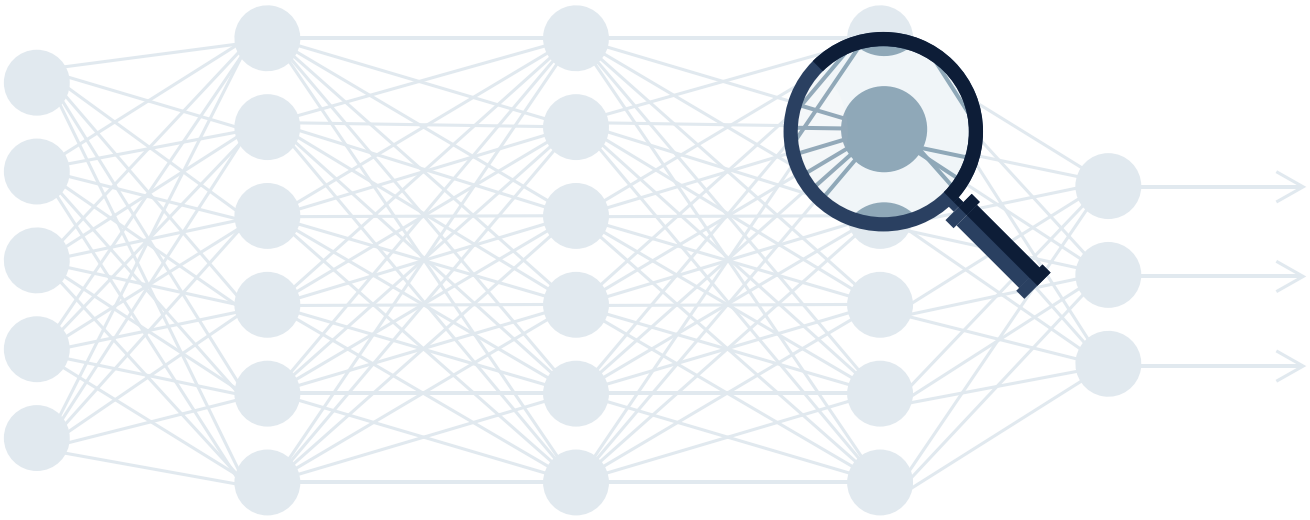
2025 JFrog Ltd. All rights reserved.

Verfügt Ihr Unternehmen über eine Lösung zum Scannen von ML-Modell-Artefakten auf Sicherheitsschwachstellen oder bösartige Modelle?
(Auftragsstudie, 2024)

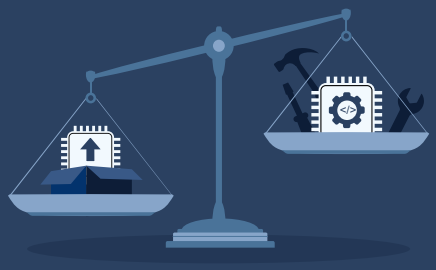
Eine deutliche Mehrheit (79 %) der Unternehmen gibt an, über eine Möglichkeit zum Scannen von ML-Modell-Artefakten auf Sicherheitsschwachstellen oder bösartigen Code zu verfügen. 11 % haben keine entsprechenden Maßnahmen im Einsatz.

Erfreulicherweise geben nur 3 % der Befragten an, selbst gehostete OSS-Modelle ohne jeglichen Scanning-Mechanismus zur Vermeidung von Schwachstellen oder bösartigen Modellen zu verwenden. Dennoch besteht weiterhin Handlungsbedarf – sowohl auf Unternehmensebene als auch innerhalb der Sicherheitsbranche –, um sicherzustellen, dass die richtigen Tools und Richtlinien für eine sichere KI-/ML-Entwicklung vorliegen.





Wichtigste Erkenntnisse

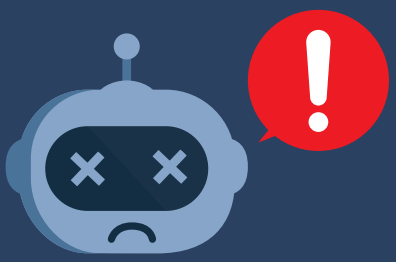


KI-Integration mit kommerziellen Modellen vorantreiben

Der Einsatz von kommerziellen Modellen scheint ein beliebter Weg zu sein, um KI-Services schneller in Unternehmensanwendungen zu integrieren. Der Zugriff auf kommerzielle Modelle per API spart Unternehmen zudem Zeit und Geld bei der Anschaffung von Tools und Ressourcen sowie dem Erwerb von Know-how, um Modelle selbst zu entwickeln und zu verwalten. Gerade für Unternehmen, die wenig Erfahrung mit KI/ML haben, kann es außerdem sinnvoll sein, die Sicherheit ihrer Modelle spezialisierten Anbietern zu überlassen, die über mehr Expertise auf diesem Gebiet verfügen.

Fehlende Transparenz in der KI-Lieferkette

Viele Unternehmen tun sich schwer damit, zuverlässige Methoden für den Umgang mit Machine-Learning-Modellen in ihren Anwendungen zu etablieren. Häufig fehlt ihnen auch eine “Single Source of Truth” für alle Softwarekomponenten, einschließlich der ML-Modelle.
Ein weiterer großer blinder Fleck besteht in der unzureichenden Nachverfolgbarkeit von Open-Source-Paketen, die transitive Abhängigkeiten im Zusammenhang mit ML-Modelle enthalten. Solche kritischen Lücken im ML-Softwareentwicklungsprozess erschweren nicht nur die effiziente Verwaltung von KI/ML-Lieferketten, sondern erhöhen auch das Risiko von Sicherheitslücken.



Übermäßiges Vertrauen in die KI-Sicherheit

Zwar berichten 79 % der Unternehmen, dass sie Modelle in gewissem Umfang scannen, doch die derzeit verfügbaren Lösungen für die Sicherheit von KI/ML-Modellen stecken mit unbedarften Erkennungsmethoden noch in den Kinderschuhen. So führten aktuelle Ansätze beispielsweise zu einer False-Positive-Rate von 96 % bei der Identifizierung bösartiger Modelle auf Hugging Face, während gleichzeitig echte Bedrohungen aufgrund von technisch noch sehr rudimentären Scanning-Verfahren übersehen wurden. Auch wenn viele Security-Tools versuchen, aus der Abdeckung von Modell-Artefakten Kapital zu schlagen, sollten Unternehmen die Effektivität dieser Modellsicherheitslösungen von Sicherheitsanbietern sorgfältig evaluieren.

2025 JFrog Ltd. All rights reserved.

Methodik

Dieser Report enthält eine Kombination von Infomationen aus JFrog-Nutzungsdaten, CVE-Analyseergebnissen des JFrog-Security-Research-Teams sowie von Drittanbietern erhobenen Umfragedaten. Es folgt eine detailliertere Übersicht über die einzelnen Quellen:

Nutzungsdaten der JFrog Plattform

Die Technologie-Nutzungstrends, die in diesem Report beleuchtet werden, stammen aus einer Momentaufnahme von anonymisierten, zum Jahresende erhobenen Nutzungsdaten der JFrog Plattform für Cloud, die Tausende von Kunden, Hunderttausende von Repositorys und Petabytes von Daten repräsentieren.

Die Beliebtheit von Paketen wird durch die Anzahl der Aktionen (Upload/Download), die Gesamtzahl der Artefakte, die Gesamtzahl der Repositorys und die Gesamtgröße der Artefakte für einen bestimmten Pakettyp dargestellt. Die Anzahl der Aktionen gibt einen guten Überblick darüber, wie oft verschiedene Pakettypen von Developern aufgerufen und generiert werden, und ist damit ein Hinweis auf die tatsächliche Nutzung in der Softwareentwicklung.

Es ist möglich, dass einige wenige Unternehmen diese Rankings verzerren. Da wir jedoch auch Artefakt-Aktionen betrachten, können wir mit Sicherheit feststellen, welcher Pakettyp aktiv im Entwicklungsprozess verwendet wird.

2025 JFrog Ltd. All rights reserved.

Analyse des JFrog-Security-Forschungsteams

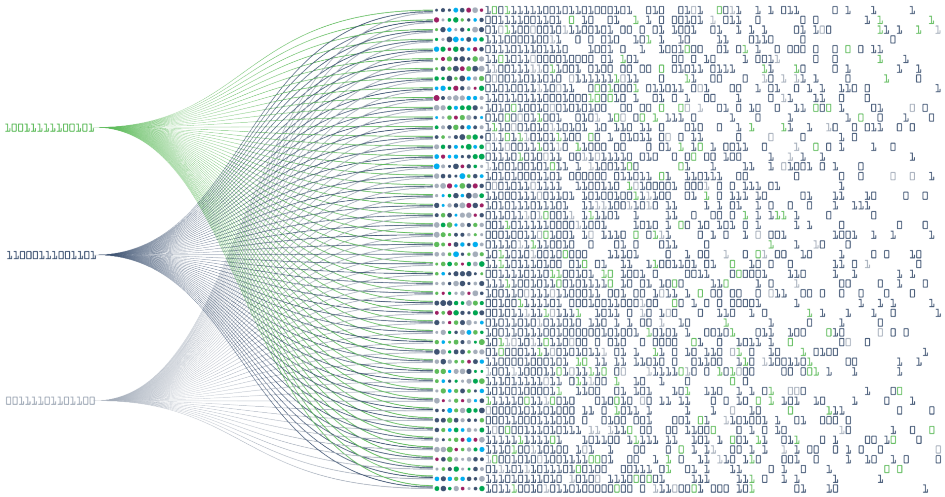
Als designierter CNA überwacht und untersucht das JFrog-Security-Forschungsteam regelmäßig neue Schwachstellen, um deren tatsächlichen Schweregrad zu ermitteln. Diese Informationen werden anschließend zum Nutzen der Community und aller JFrog-Kunden veröffentlicht.

Dieser Report enthält Daten, die aus öffentlichen Quellen über den JFrog Catalog Service gewonnen wurden, CVE-Informationen aus der National-Vulnerability-Database und eigene Analysen des JFrog-Security-Research-Teams auf Basis dieser Datenquellen.

In Auftrag gegebene Umfrageergebnisse

JFrog hat Atomik Research mit der Durchführung einer internationalen Online-Umfrage unter 1.402 Befragten aus ausgewählten Branchen[1], die in den USA (n=375), Großbritannien (n=205), Indien (n=206), Deutschland (n=205), Frankreich (n=205) und Israel (n=205) tätig sind, beauftragt. Die Stichprobe setzt sich aus Vollzeitbeschäftigten zusammen, die bestimmte Funktionen[2] in der IT-, Informationssystem- oder Technologieabteilung ihres Unternehmens ausüben. Darüber hinaus gaben alle Befragten an, dass ihr Unternehmen insgesamt 1.000 oder mehr Mitarbeiter beschäftigt, und bestätigten, dass es in ihrem Unternehmen ein Softwareentwicklungsteam mit mindestens 50 Teammitgliedern gibt. Alle Teilnehmer hatten die Möglichkeit, den Online-Fragebogen in Englisch, Französisch, Deutsch, Hebräisch oder Hindi auszufüllen.

Die Fehlerquote für die Gesamtstichprobe beträgt +/- 3 Prozentpunkte bei einem Konfidenzniveau von 95 Prozent. Die Feldforschung fand zwischen dem 22. November und dem 9. Dezember 2024 statt. Atomik Research ist eine kreative Marktforschungsagentur.



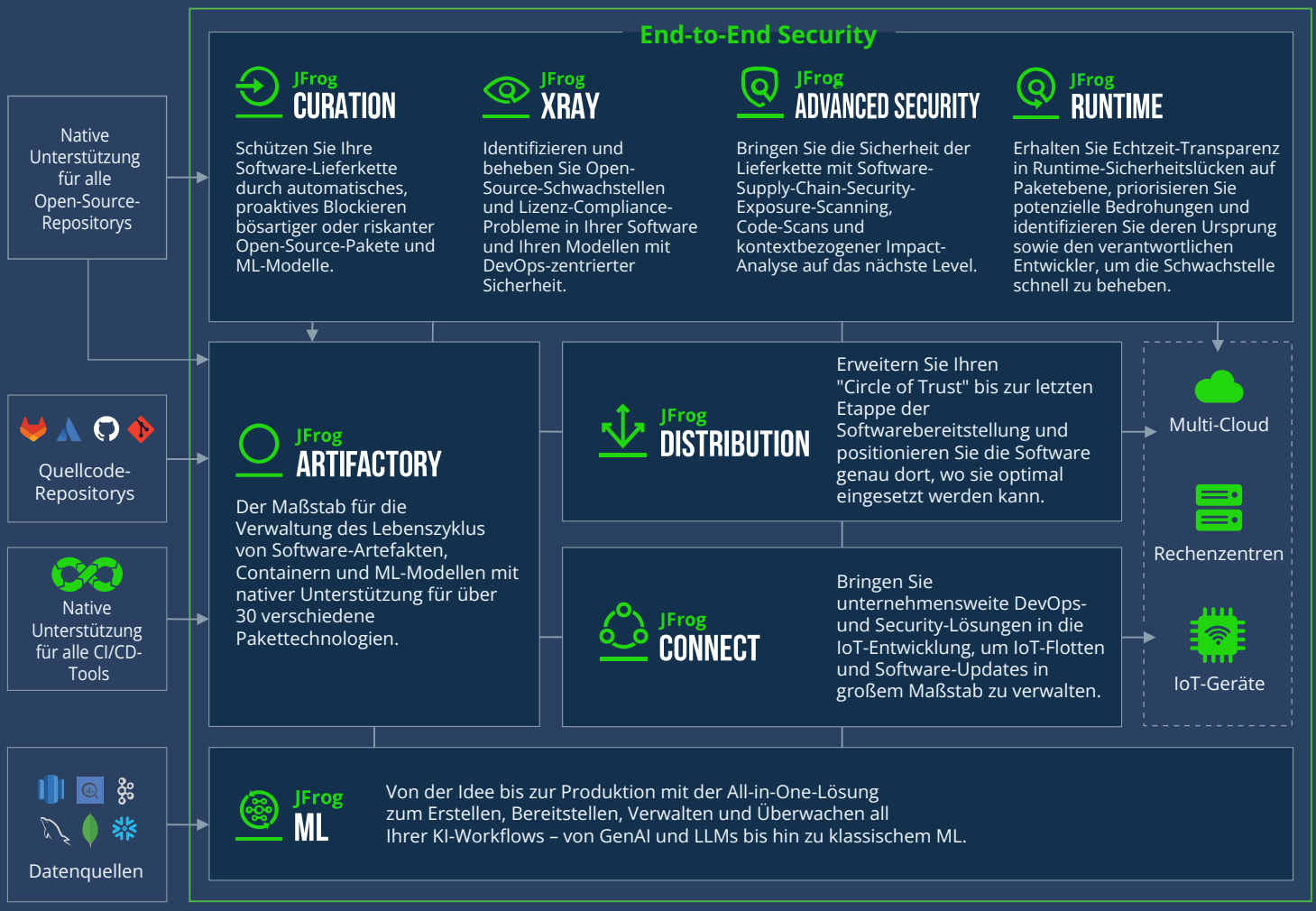
[1]Zur Teilnahme waren nur Befragte qualifiziert, die angaben, dass sie in einem Unternehmen beschäftigt sind, das in einer der folgenden Branchen aktiv ist: (a.) Luft- und Raumfahrt (b.) Architektur und Ingenieurwesen (c.) Automobilindustrie (d.) Banken, Finanzdienstleistungen, Versicherungen & Fintech (e.) Energie, Öl, Gas (f.) Regierung oder öffentlicher Sektor (g.) Gesundheitswesen und Life Sciences (h.) Gastgewerbe (i.) Fertigungsindustrie (j.) Einzelhandel (k.) Technologie (l.) Transport und Logistik (m.) Versorgungsunternehmen, Telekommunikation & Energieversorgung

[2]Zur Teilnahme waren nur Befragte qualifiziert, die angaben, dass sie eine der folgenden oder vergleichbare berufliche Funktionen ausüben: (a.) KI-Spezialist oder KI-Engineer (b.) Application Security Engineer (d.) Cybersecurity Engineer (e.) Data Scientist (f.) Entwickler (g.) DevOps-Architekt (h.) DevOps-Engineer (i.) Engineering Manager (j.) Machine-Learning-Spezialist oder ML-Engineer (k.) Platform Engineer (l.) Sicherheitsarchitekt (m.) Security Researcher (n.) Site Reliability Engineer (o.) Softwarearchitekt (p.) Softwareentwickler (q.) Software Engineer (r.) Solution Architect. Zusätzlich mussten die Teilnehmenden angeben, in der Informationstechnologie-, Informationssystem-, Technologie- oder IT-Produktentwicklungsabteilung ihres Unternehmens tätig zu sein.

2025 JFrog Ltd. All rights reserved.

Über die JFrog Plattform

Die JFrog Plattform ist eine hochskalierbare, offene, Cloud-native Lösung, die sich in die Pakettechnologien und Tools der Software-Lieferkette integrieren lässt. Sie bietet Unternehmen volle Kontrolle und Traceability, während Softwarekomponenten vom Entwickler in alle Deployment-Umgebungen gelangen, einschließlich ML-Modelle, Edge-Geräte und Produktionsrechenzentren.



Dieser Daten-Report enthält "zukunftsgerichtete" Aussagen im Sinne der US-amerikanischen Bundeswertpapiergesetze, einschließlich, aber nicht beschränkt auf Aussagen über die JFrog-Nutzungsdaten und die Software-Lieferkette.

Diese zukunftsgerichteten Aussagen basieren auf unseren derzeitigen Annahmen, Erwartungen und Überzeugungen und unterliegen erheblichen Risiken, Ungewissheiten, Annahmen und veränderten Umständen, die dazu führen können, dass die tatsächlichen Ergebnisse, Leistungen oder Errungenschaften der Produkte von JFrog wesentlich von denen abweichen können, die in den zukunftsgerichteten Aussagen ausgedrückt oder impliziert werden. Es gibt eine beträchtliche Anzahl von Faktoren, die dazu führen könnten, dass die tatsächlichen Ergebnisse, Leistungen oder Errungenschaften wesentlich von den in diesem Report gemachten Aussagen abweichen, einschließlich, aber nicht beschränkt auf die Risiken, die in unseren Unterlagen bei der Securities and Exchange Commission aufgeführt sind, einschließlich unseres Jahresberichts auf Formblatt 10-K für das am 31. Dezember 2024 zu Ende gegangene Jahr, unserer Quartalsberichte auf Formblatt 10-Q und anderer Unterlagen und Berichte, die wir von Zeit zu Zeit bei der Securities and Exchange Commission einreichen können. Zukunftsgerichtete Aussagen spiegeln nur unsere Überzeugungen und Annahmen zum Zeitpunkt dieses Reports wider. Wir lehnen jede Verpflichtung ab, zukunftsgerichtete Aussagen zu aktualisieren.

2025 JFrog Ltd. All rights reserved.